LILAW INC., A LAW CORPORATION
J. James Li, Ph.D. (SBN 202855, lij@lilaw.us)
Tony Abdollahi (SBN 157235, abdollahia@lilaw.us)
5050 El Camino Real, Suite 200
Los Altos, California 94022
Telephone: (650) 521-5956
Facsimile: (650) 521-5955

Attorneys for Plaintiff TS Patents LLC

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

TS PATENTS LLC, a California limited liability company,

Plaintiff,

vs.

YAHOO! INC., a Delaware corporation, and DOES 1-50, inclusive,

Defendants.

COMPLAINT FOR PATENT INFRINGEMENT

DEMAND FOR JURY TRIAL

PARTIES AND NATURE OF ACTION

1. This is an action for patent infringement arising under the Patent Laws of the United States, 35 U.S.C. § 1, *et seq.*, as alleged by Plaintiff TS Patents LLC ("TS Patents") against defendant Yahoo! Inc. ("Yahoo") and defendants Does 1 through 50. Yahoo and Does 1-50 are collectively referred to as the "Defendants" hereinafter.

2. TS Patents is a California limited liability company with its registered office located in Fremont, California 94568.

3. On information and belief, Yahoo is incorporated in Delaware and headquartered at 701 First Ave Sunnyvale, CA 94089.

4. Defendants Does 1 through 50 include Yahoo's customers and licensees who have used Yahoo's infringing products and/or services. Defendants Does 1 through 50 may also include distributors, contractors, and other individuals and entities who have acted on behalf of Yahoo in making, using, selling, and/or offering for sale Yahoo's infringing products or services. The identities of Does 1 through 50 will be revealed through discovery.

JURISDICTION AND VENUE

5. This Court has personal jurisdiction over Defendant Yahoo because it has its corporate headquarters in Sunnyvale, California.

6. This Court has the original subject matter jurisdiction over this action pursuant to 35 U.S.C. § 100 *et seq.*, and 28 U.S.C. § 1338(a).

7. Venue is proper in the Northern District of California pursuant to (i) 28 U.S.C. § 1391(c)(2) because Yahoo is subject to personal jurisdiction and maintains its principal place of business in this judicial district, and (ii) 28 U.S.C. § 1400(b) because Yahoo has committed acts of infringement and has a regular and established place of business in this judicial district.

GENERAL ALLEGATIONS

8. TS Patents is owned and operated by Sheng Tai "Ted" Tsao ("Tsao"), an individual inventor and entrepreneur. Tsao is the inventor of multiple U.S. patents and pending patent applications, which are collectively referred to as the "Tsao patents" in this Complaint. As of the filing date of the Complaint, there are 40 issued and active Tsao Patents that are classified into five families, as shown in Exhibit A. Tsao initially founded a company, STT WebOS, Inc., to develop products incorporating his inventions. Recently, Tsao formed TS Patents for the purpose of licensing and enforcing his inventions featured in the Tsao patents.

9. Founded on or about 1995 by Jerry Yang and David Filo, Yahoo is an internet portal and search engine. Yahoo owns Tumblr, a microblogging and social networking website founded by David Karp in 2007. Tumblr allows users to post multimedia and other content to a short-form blog.

10. Mr. Tsao presented his patented technology to Yahoo initially in 2004 and, since 2016, has been trying to communicate with Yahoo about potential licensing, but has been unable to achieve a resolution of these claims.

a. In 2004 and 2005, Ms. Tsao communicated with two Yahoo managers to persuade Yahoo to license the advanced online folder software, which are implements of some of the Tsao Patents, to enhance Yahoo's Briefcase service.

b. Specifically, on August 16, 2004, Mr. Tsao emailed Yahoo manager Toby Coppel, Senior VP Corporate Development, and referring to a September 9, 2003 meeting, noted, "after that [meeting] I have sent you our STT WebOS technology. Now, I am pleased to let you know that we have developed a new brief-case service software, which could be perfectly used by Yahoo for its briefcase service on web."

c. Then, on March 31, 2005, Mr. Tsao emailed Yahoo manager, Steve Kohler, Sr. Technical Acquisition Partner, and advised, "I believe that our GSSFS or our data mgmt technology may immediately add values to Yahoo's business ... if Yahoo has developed an interest in trying GSSFS and our data mgmt technology, we can arrange a meeting with you or your people either at Yahoo or at our office."

d. Although Yahoo declined these offers, Ms. Tsao later learned that Yahoo had, in fact, adopted the technology reflected in the patents.

e. Accordingly, on June 18, 2016, Mr. Tsao sent a letter to Marissa Mayer and Ron Bell of Yahoo advising, "we have concluded that Yahoo has been practicing the inventions claimed by the Information-Sharing patents and Web-Folder patents. Specially, in 2004 STT WebOS, Inc, had introduced the web folder technology to Yahoo. This letter thus serves as notice of Yahoo's exemplary infringement of the Information-Sharing patents and Web-Folder patents …" Mr. Tsao offered to license the Tsao Patents to Yahoo

f. On August 1, 2016, a Senior Legal Director of Yahoo replied to Mr. Tsao's letter requesting more specificity regarding the alleged infringement of the Tsao Patents by Yahoo.

g. On September 30, 2016, the undersigned law firm sent Yahoo three preliminary claim charts showing how Yahoo practices the Tsao Patents, including the '442, '473, and '547 Patents; later a fourth claim chart was sent to Yahoo for the '891 patent.

h. On February 10, 2017, Yahoo's outside litigation counsel, Morgan, Lewis & Bockius LLP, wrote to TS Patents' counsel and asserted that Yahoo did not infringe the Patents-in-suit.

i. On February 21, 2017, the undersigned law firm wrote back to Yahoo's outside counsel pointing out why the non-infringement arguments made by Yahoo had no merits.

j. Later, Yahoo's outside counsel stated to TS Patents by telephone that Yahoo had no authority to negotiate the licensing terms proposed by TS Patents, because of the pending acquisition of Yahoo by Verizon. Consequently, TS Patents' licensing attempt with Yahoo hit an impasse. The instant lawsuit ensued.

11. TS Patents considers it likely that Yahoo has infringed, and/or is infringing, all five families of the Tsao Patents. However, due to the limitation of publicly available evidence, TS

Patents has been able to collect infringing evidence on only 3 families: the Web Multitasking Family, the Web Folder Family, and the Information Sharing Family. Of the 3 families, TS Patents asserts four patents against Yahoo's email and Tumblr services: U.S. Patent Nos. 8,396,891 (the "'891 Patent"), 8,713,442 (the "'442 Patent'"), 8,799,473 (the "'473 Patent") and 9,280,547 (the "'547 Patent"), which are collectedly referred to as the "Patents-in-Suit." Exhibits B, C, D, and E are true and correct copies of the Patents-in-Suit. TS Patents reserves the right to add more patents to the Patents-in-Suit and/or replace one or more Patents-in-Suit with similar patents in the same families of the Tsao Patents.

12. TS Patents is the current assignee and owner of all of the Tsao Patents, including the Patents-in-Suit.

FIRST CAUSE OF ACTION

Infringement of U.S. Patent No. 8,396,891

13. TS Patents hereby incorporates by reference the foregoing averments as if fully set forth herein.

14. The '891 Patent belongs to the Information Sharing Family of the Tsao Patents.

15. Yahoo's Tumblr practices one or more claims of the '891 patent, as shown by the illustrative claim chart in Exhibit F.

16. Yahoo's infringement of the '891 Patent is willful, as evidenced by the lack of any meritorious defenses asserted by Yahoo during the parties' negotiation.

17. Defendants' infringement of the '891 Patent has caused significant damages to TS Patents in terms of lost royalties.

SECOND CAUSE OF ACTION

Infringement of U.S. Patent No. 8,713,442

18. TS Patents hereby incorporates by reference the foregoing averments as if fully set forth herein.

19. The '442 Patent belongs to the Information Sharing Family of the Tsao Patents.

20. Yahoo's Tumblr practices one or more claims of the '442 patent, as shown by the illustrative claim chart in Exhibit G.

21. Yahoo's infringement of the '442 Patent is willful, as evidenced by the lack of any meritorious defenses asserted by Yahoo during the parties' negotiation.

22. Defendants' infringement of the '442 Patent has caused significant damages to TS Patents in terms of lost royalties.

THIRD CAUSE OF ACTION

Infringement of U.S. Patent No. 8,799,473

23. TS Patents hereby incorporates by reference the foregoing averments as if fully set forth herein.

24. The '473 Patent belongs to the Web Multitasking Family of the Tsao Patents.

25. Yahoo's Tumblr practices one or more claims of the '473 Patent, as shown by the illustrative claim chart in Exhibit H.

26. Yahoo's email service, Yahoo! Mail, also practices one or more claims of the '473 Patent, as shown by the illustrative claim chart in Exhibit I.

27. Yahoo's infringement of the '473 Patent is willful, as evidenced by the lack of any meritorious defenses asserted by Yahoo during the parties' negotiation.

28. Defendants' infringement of the '473 Patent has caused significant damages to TS Patents in terms of lost royalties.

FOURTH CAUSE OF ACTION

Infringement of U.S. Patent No. 9,280,547

29. TS Patents hereby incorporates by reference the foregoing averments as if fully set forth herein.

30. The '547 Patent belongs to the Web Folder Family of the Tsao Patents.

31. Yahoo's Tumblr practices one or more claims of the '547 patent, as shown by the illustrative claim chart in Exhibit J.

32. Yahoo's infringement of the '547 Patent is willful, as evidenced by the lack of any meritorious defenses asserted by Yahoo during the parties' negotiation.

33. Defendants' infringement of the '547 Patent has caused significant damages to TS Patents in terms of lost royalties.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff prays for relief against Defendants as follows:

a. The Court shall grant permanent injunction against Defendants, their agents, attorneys, representatives, and those acting in concert with Defendants, pursuant to 35 U.S.C. § 283;

b. The Court shall award compensatory damages according to the proof, including treble damages under 35 U.S.C. § 284;

c. The Court shall declare the case as an exceptional case under 35 U.S.C. § 285, and award Plaintiff reasonable attorney's fees and costs according to the proof;

d. The Court shall award prejudgment interest at the maximum legal rate as allowed by the law;

e. The Court shall award costs of suit herein incurred; and

f. The Court shall award such other and further relief as the Court may deem proper.

JURY DEMAND

Plaintiff hereby demands a trial by jury on all issues so triable.

DATED: March 29, 2017

LILAW INC.
ATTORNEYS FOR PLAINTIFF TS PATENTS LLC

By /s/ J. James Li
J. James Li, Ph.D.

Exhibit A

Tsao Patents

1: Web Multitasking Family:

No.	Patent No.	Issue Date
1.	8,812,682	08/19/2014
2.	8,799,473	08/05/2014
3.	8,892,742	11/18/2014
4.	9,009,314	04/14/2015
5.	9,015,321	04/21/2015
6.	9,307,055	04/05/2016
7.	8,745,193	06/03/2014
8.	8,566,463	10/22/2013
9.	7,418,702	09/26/2008

2: Centralized Cloud Operations Family:

	Patent No.	Issue Date
10.	9,288,266	03/15/2016
11.	8,909,976	12/09/2014
12.	9,106,677	08/11/2015
13.	8,903,963	12/02/2014
14.	8,688,772	04/01/2014
15.	8,639,788	01/28/2014
16.	8,473,596	06/25/2013
17.	7,379,990	05/27/2008
18.	9,467,513	10/11/2016
19.	9,602,600	03/21/2017

3: Web Folder Family:

	Patent No.	Issue Date
20.	9,323,757	04/26/2016
21.	9,317,510	04/19/2016
22.	9,280,547	03/08/2016
23.	9,146,932	09/29/2015
24.	9,390,094	07/12/2016
25.	8,341,258	12/25/2012
26.	8,589,527	11/19/2013
27.	8,812,640	09/09/2014
28.	7,945,652	05/17/2011

4: Information Sharing Family:

	Patent No.	Issue Date
29.	8,713,442	04/29/2014
30.	8,396,891	03/12/2013
31.	8,577,839	11/05/2013
32.	8,977,722	03/10/2013
33.	9,448,697	09/20/2016

5: Web Storage Family:

	Patent No.	Issue Date
34.	9,219,780	12/22/2015
35.	9,239,686	01/19/2016
36.	9,098,526	08/04/2015
37.	8,868,690	10/21/2014
38.	8,856,195	10/07/2014
39.	8,606,880	12/10/2013
40.	9,201,961	12/01/2015

Exhibit B

US008396891B2

(12) **United States Patent**
Tsao

(10) **Patent No.: US 8,396,891 B2**
(45) **Date of Patent: Mar. 12, 2013**

(54) **METHOD AND APPARATUS OF DYNAMIC UPDATING WEB PORTALS**

(75) Inventor: **Sheng Tai (Ted) Tsao**, San Jose, CA (US)

(73) Assignee: **STT Web OS, Inc.**, San Jose, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **13/326,178**

(22) Filed: **Dec. 14, 2011**

(65) **Prior Publication Data**

US 2012/0089589 A1 Apr. 12, 2012

Related U.S. Application Data

(63) Continuation of application No. 12/511,039, filed on Jul. 28, 2009, which is a continuation-in-part of application No. 11/732,496, filed on Apr. 2, 2007, which is a continuation-in-part of application No. 10/713,904, filed on Aug. 6, 2002, now Pat. No. 7,418,702, and a continuation-in-part of application No. 10/713,905, filed on Aug. 12, 2002, now Pat. No. 7,379,990, said application No. 12/511,039 is a continuation-in-part of application No. 11/374,302, filed on Jul. 2, 2004, now Pat. No. 7,945,652, which is a continuation-in-part of application No. 10/713,904, filed on Aug. 6, 2002, now Pat. No. 7,418,702, and a continuation-in-part of application No. 10/713,905, filed on Aug. 12, 2002, now Pat. No. 7,379,990.

(60) Provisional application No. 60/787,653, filed on Mar. 31, 2006.

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/781**

(58) **Field of Classification Search** ..... None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,604,494	A *	8/1986	Shepard, Jr.	136/246
2002/0129106	A1 *	9/2002	Gutfreund	709/205
2003/0097410	A1 *	5/2003	Atkins et al.	709/206
2005/0262006	A1 *	11/2005	Beartusk et al.	705/37
2006/0095514	A1 *	5/2006	Wang et al.	709/204
2009/0138808	A1 *	5/2009	Moromisato et al.	715/758

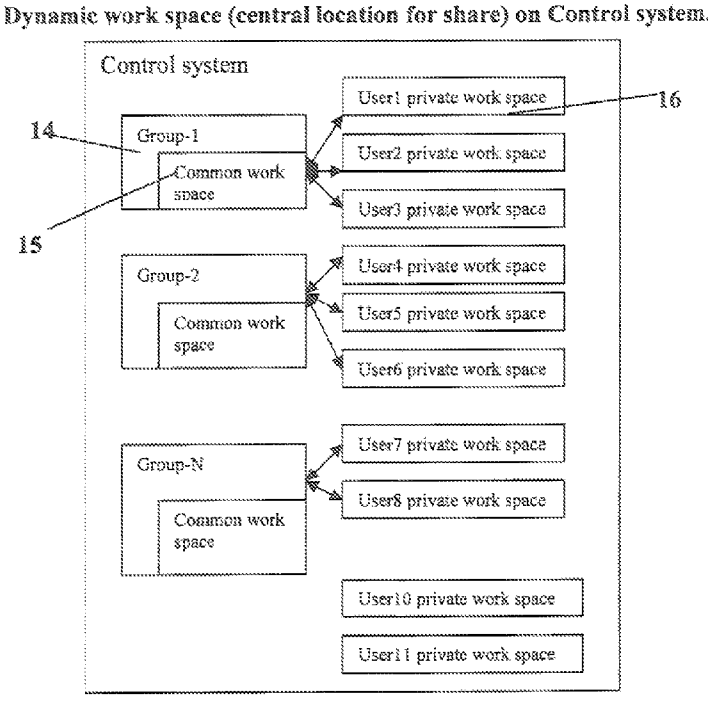
* cited by examiner

Primary Examiner — Syed Hasan
(74) *Attorney, Agent, or Firm* — Foley & Lardner LLP; Antoinette F. Konski; Alex Y. Nie

(57) **ABSTRACT**

The web portal's contents management becomes complex and sometime tedious job. There are needs for an efficient and cost effective way to manage large volumes of the web contents for web portals. The posting and withdrawing operation of the present invention has provided a means for making the tasks of updating and managing web portal's contents become much easy and efficient due to the principle of the parallelism can be deployed.

5 Claims, 18 Drawing Sheets



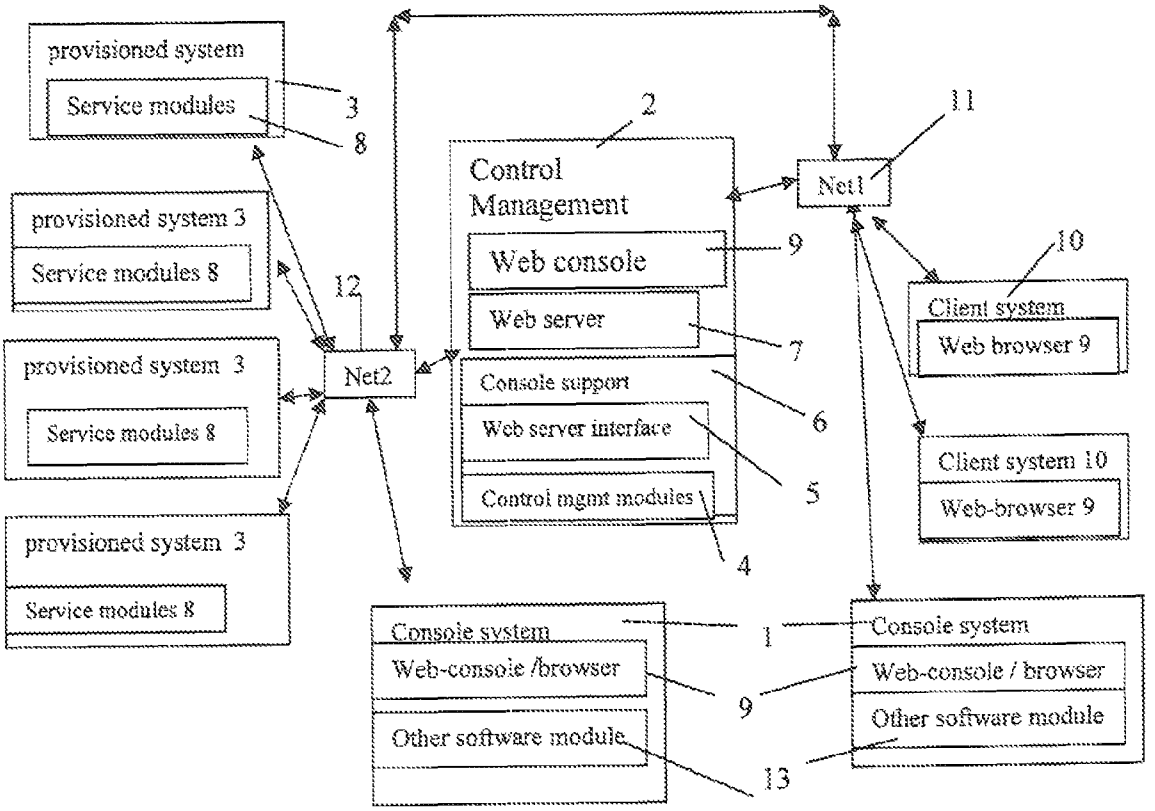

provisioned system
Service modules
3
8
provisioned system 3
Service modules 8
provisioned system 3
Service modules 8
provisioned system 3
Service modules 8
12
Net2
2
Control
Management
Web console
9
Web server
7
Console support
6
Web server interface
5
Control mgmt modules
4
11
Net1
10
Client system
Web browser 9
Client system 10
Web-browser 9
Console system
1
Web-console /browser
9
Other software module
13
Console system
Web-console / browser
Other software module

Fig. 1

Fig. 2 A variation of CCDSVM platform for web-browser based communication:

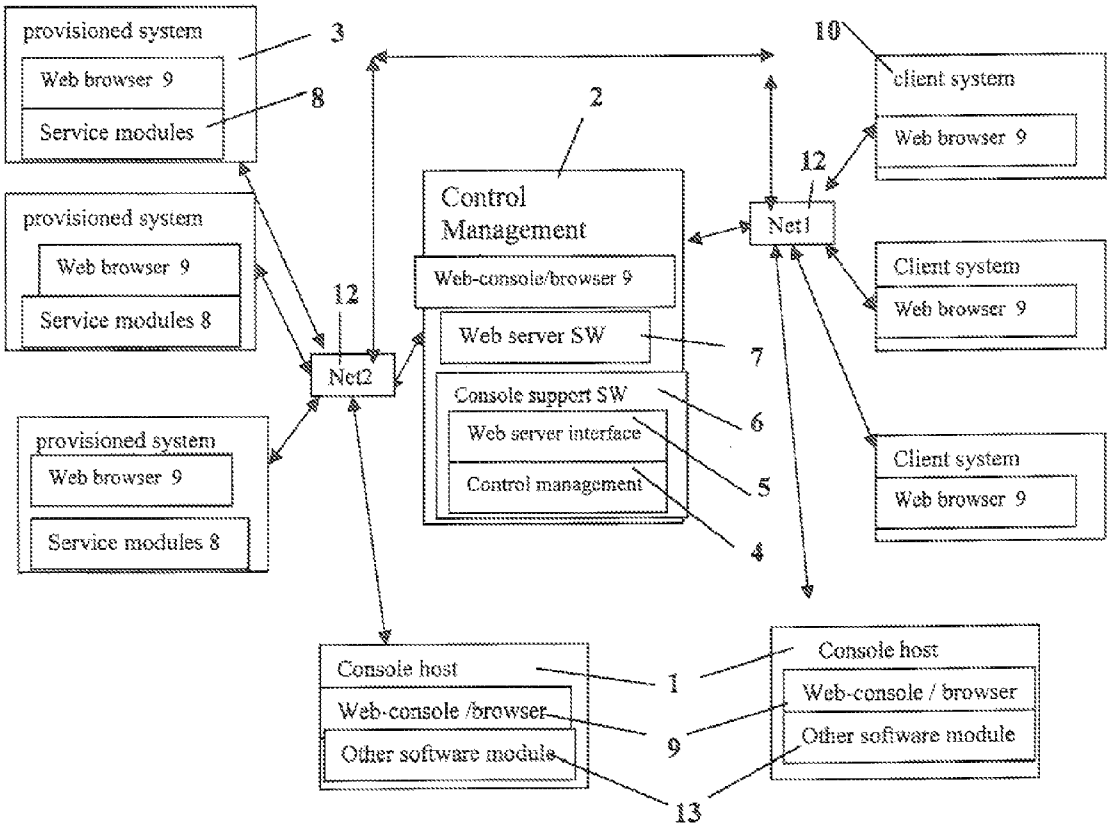


Fig. 2A

Fig. 2B: A variation of CCSDVM platform, the degenerated CCDSVM with only control system and without provisioned systems, and to be accessed by massive client systems. (The like elements has same reference number)

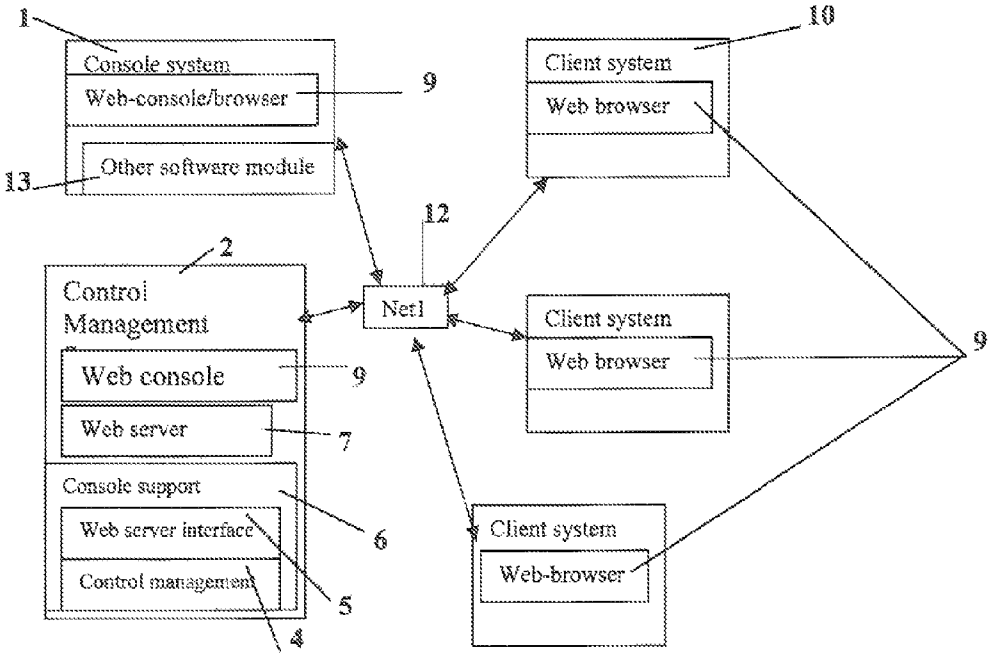


Fig. 2B

Fig. 3) The Software of WCUWE for typical CCDSVM:

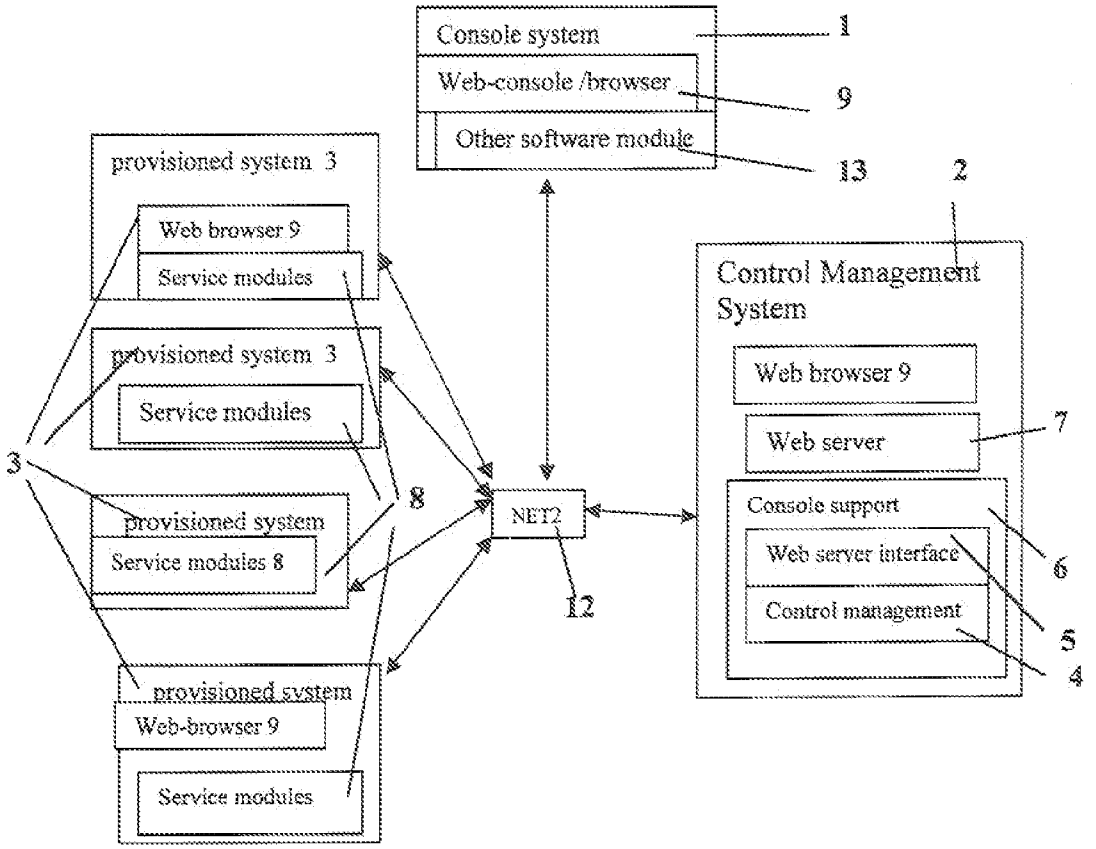


Fig. 3

Fig. 4A: Dynamic work space (central location for share) on Control system.

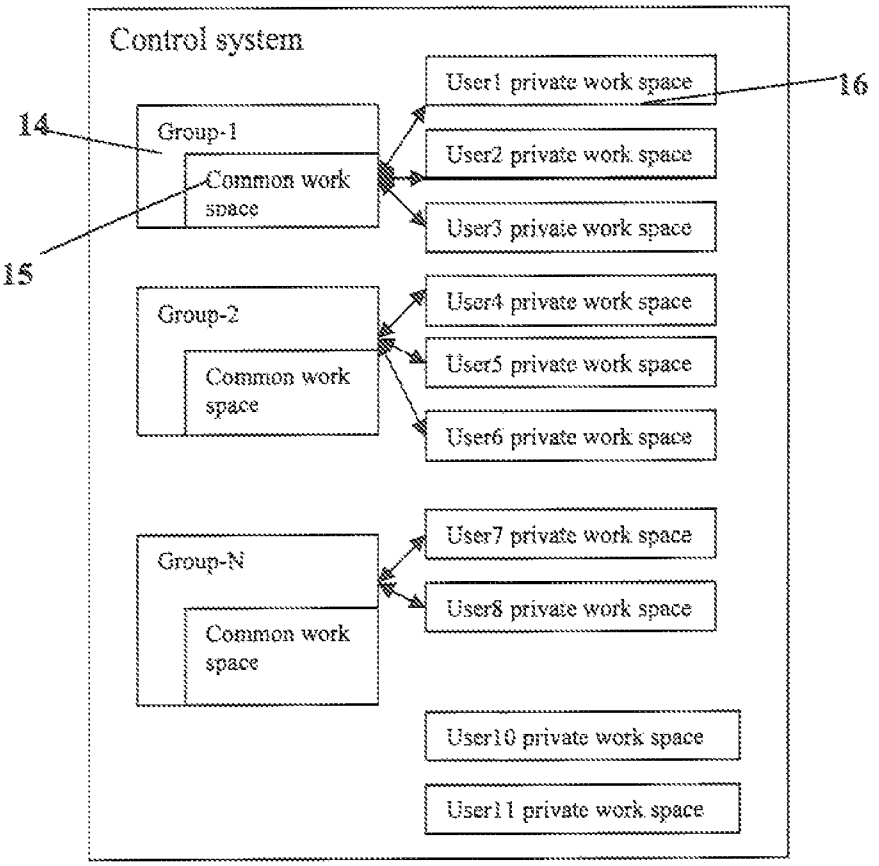


Fig. 4A

Fig. 4B: an example of resource may be assigned to a user-group common work space.

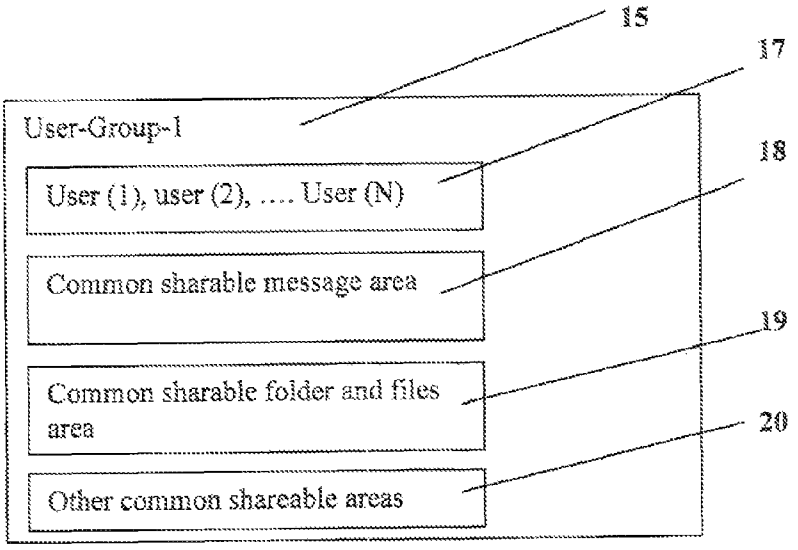


Fig. 4B

Fig. 4C: an example of user private work space.

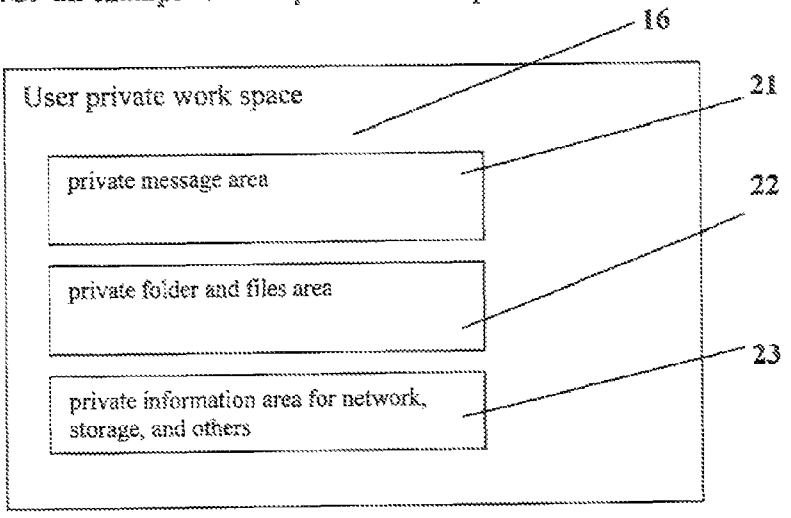


Fig. 4C

Fig. 5: An example of resource objectives on systems and can be assigned to each group or each user' work space.

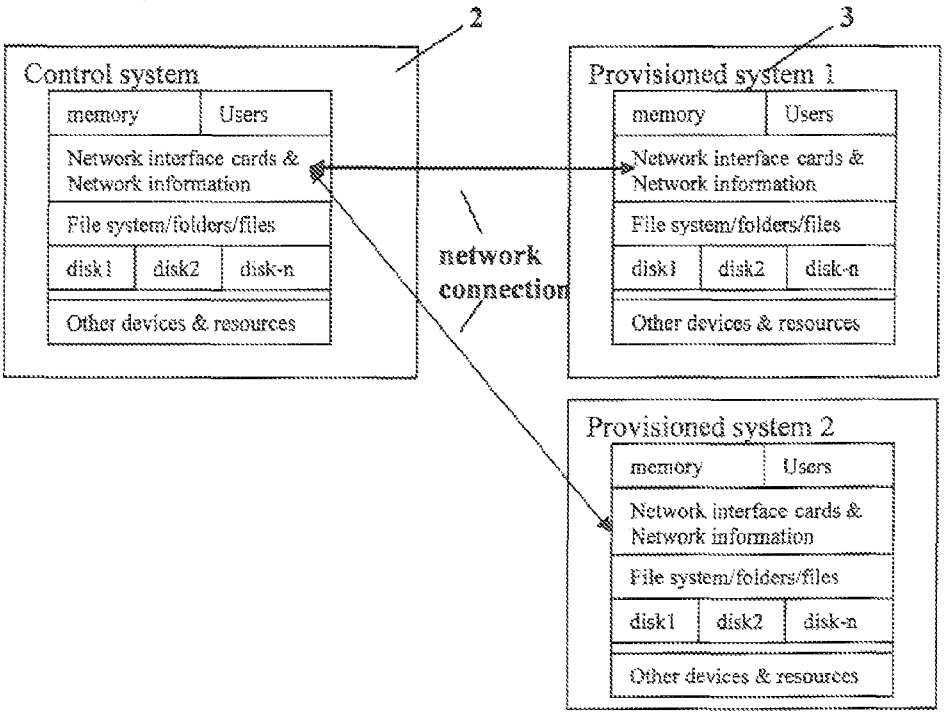


Fig. 5

Fig. 6A: An example of displaying a web-page with 4 sections in web-browser of user X in user-group-1 **during** an interactive online meeting.

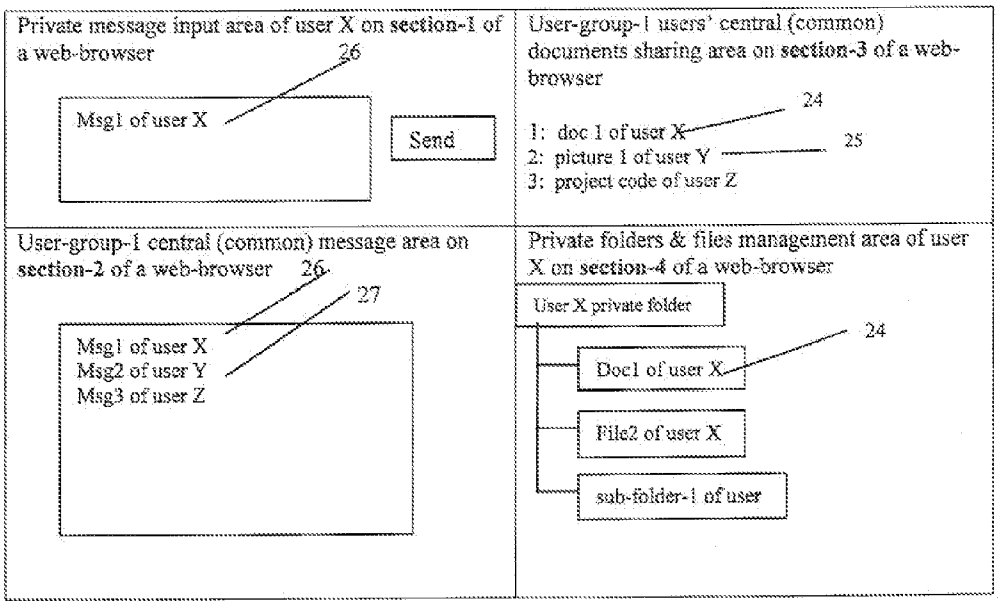
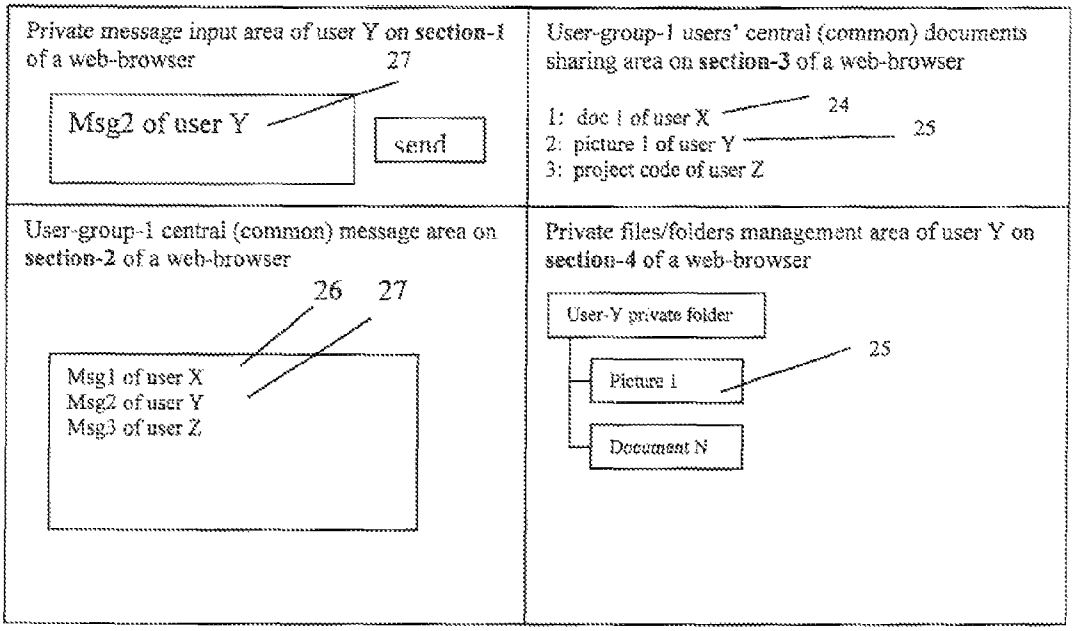


Fig. 6A

Fig. 6B: An example of displaying a web-page with 4 sections in web-browser of user Y in user-group-1 during interactive online meeting.



group-1 before interactive online meeting.

Fig. 6B

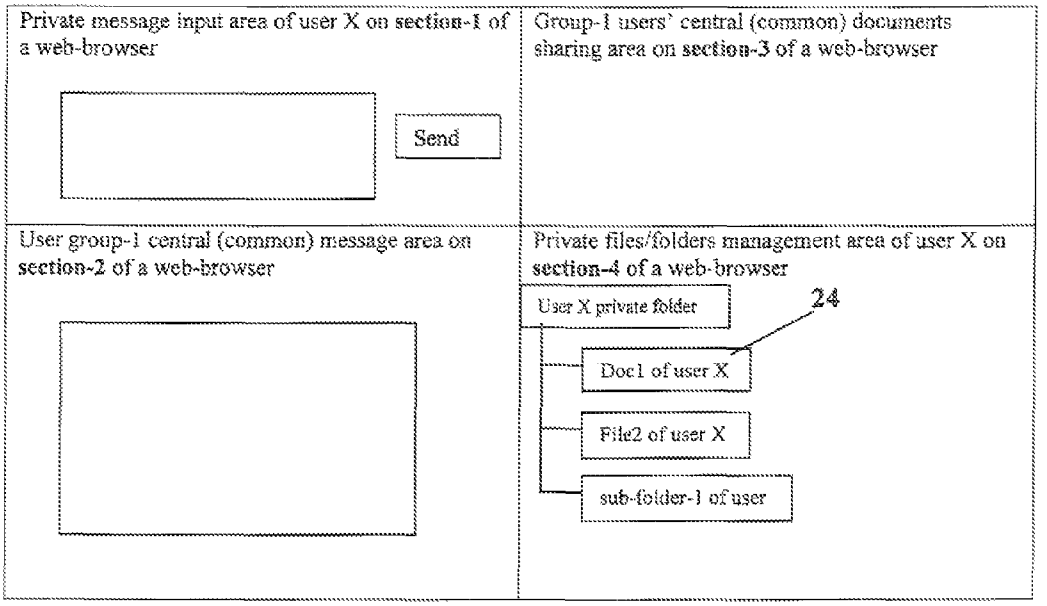


Fig. 6C

Fig. 6D: An example of display a web-page with 4 sections in browser of user Y in user group-1 before interactive online meeting.

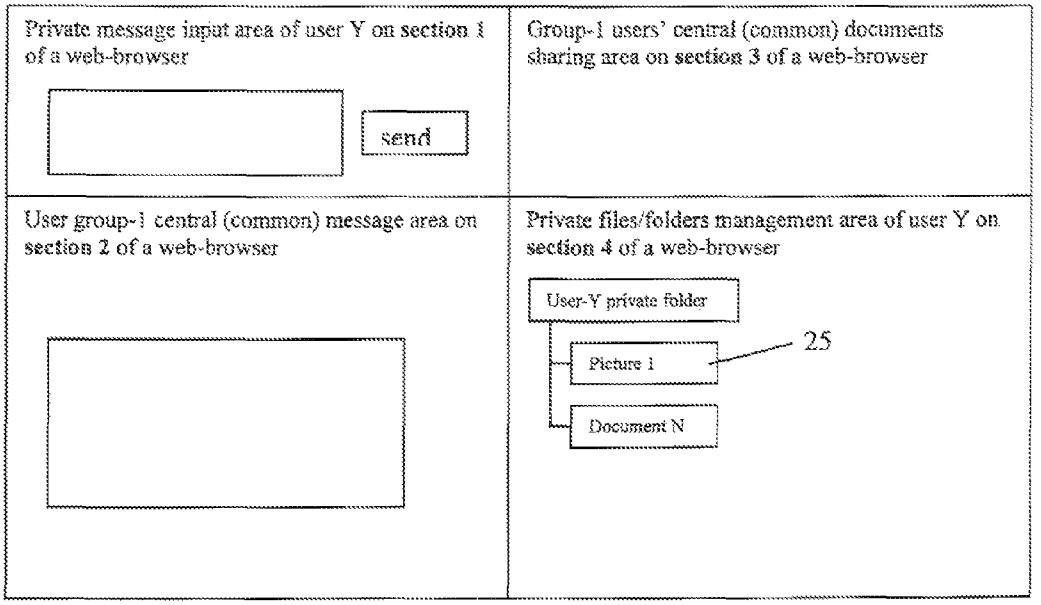


Fig. 6D

Fig. 6E: An example of displayed web-page that each massive online user can access and view from each of their browser without login to CCDSVM during said user X and user Y in user-group-1 online meeting.

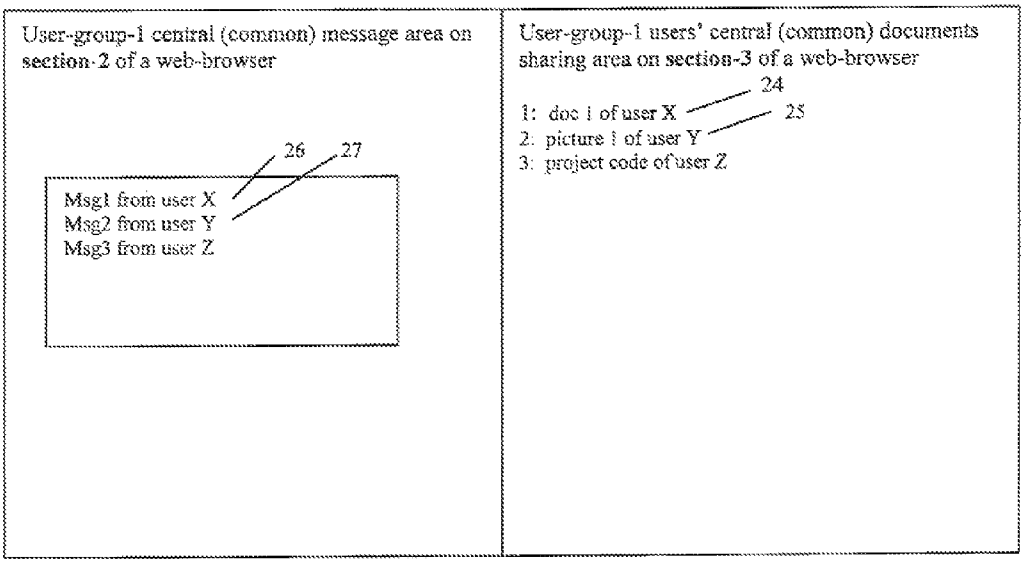


Fig. 6E

Fig. 7A: Non-group based one-to-one peer-to-peer post messages and folders of user X

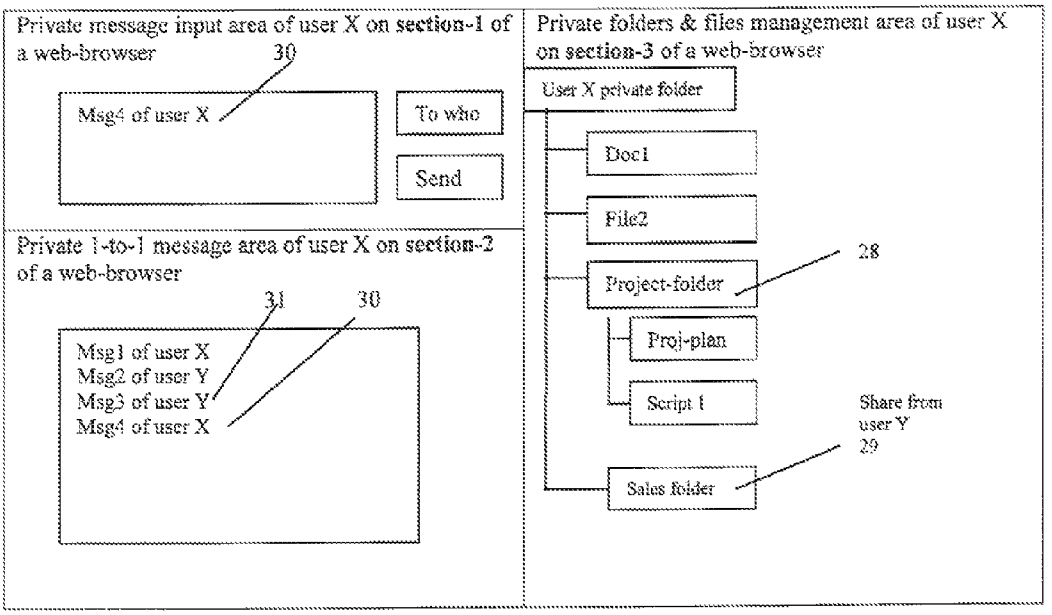


Fig. 7A

Fig. 7B: Non-group based one-to-one peer-to-peer post messages and folders of user Y

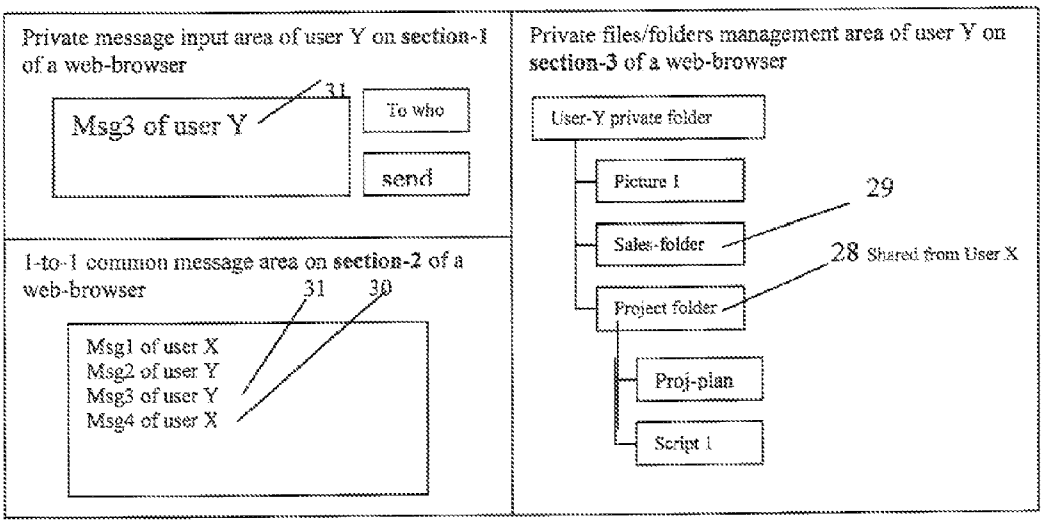


Fig.7B

Fig. 8A: An example web graphic presentation view of assigned resources in a user X's private work space.

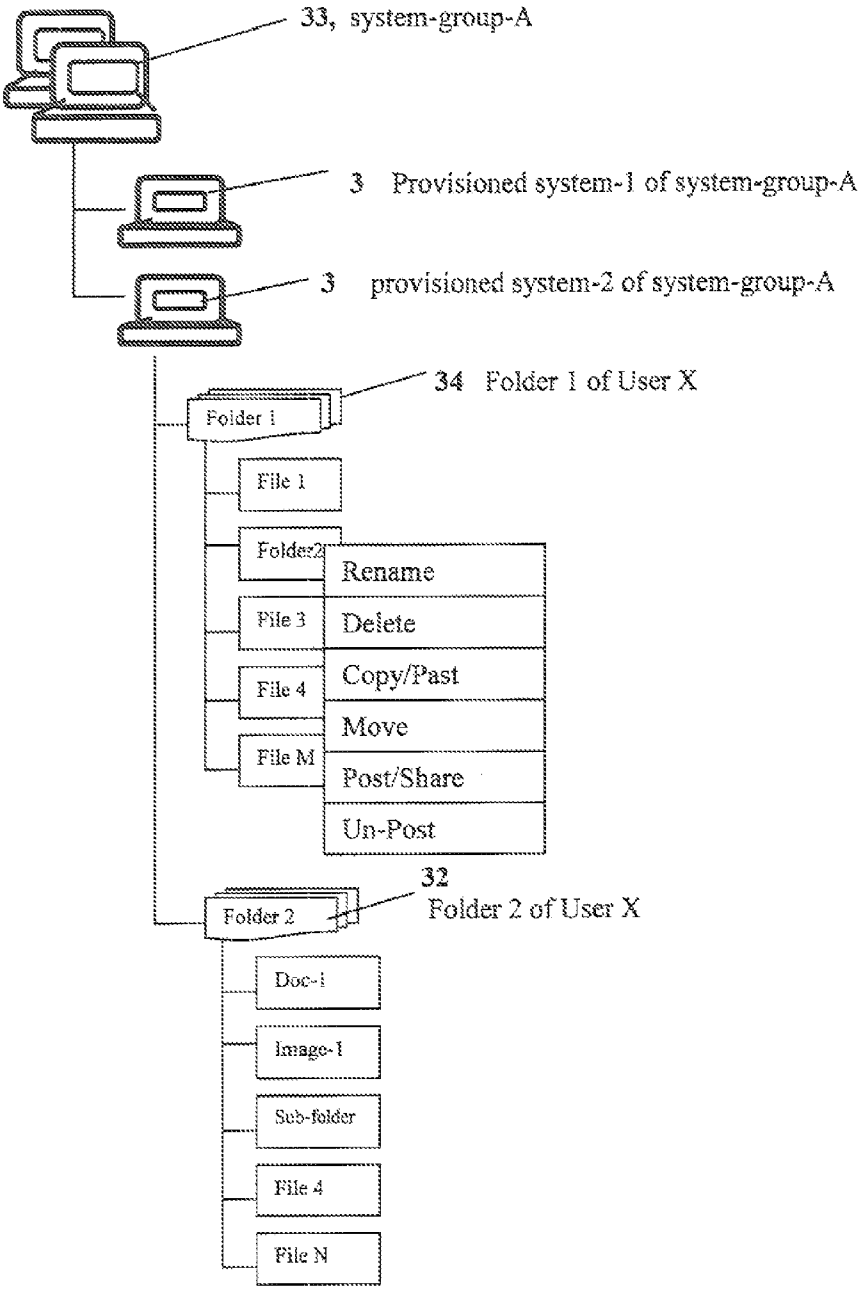


Fig. 8A

Fig. 8B examples of web-based graphic presentation of assigned resource in the private work space of user Y.

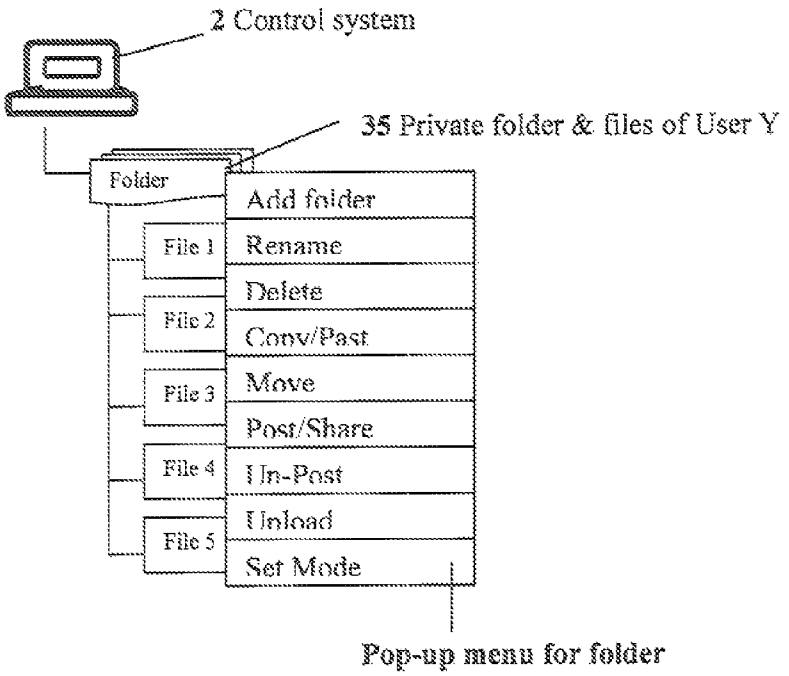


Fig.8B

Fig. 8C: An example web graphic presentation view of user Y's private work space after user X dynamically posted and shared "folder 2" in user X's private space.

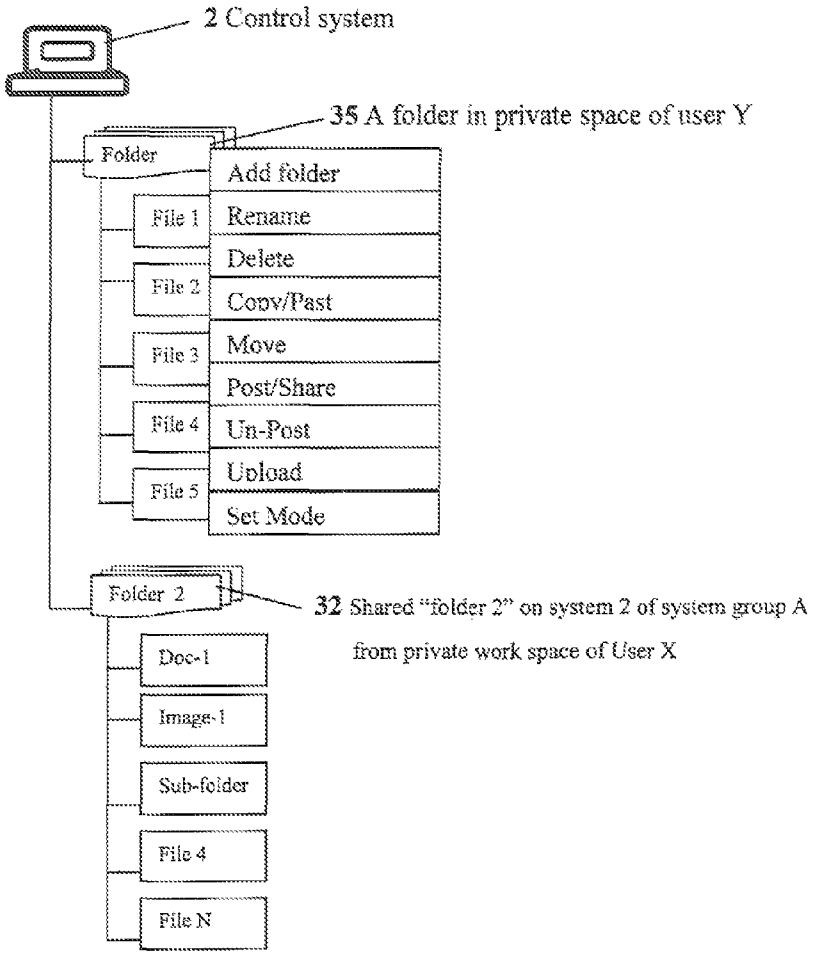


Fig. 8C

Fig. 9A: an example of pop-up menu for folder

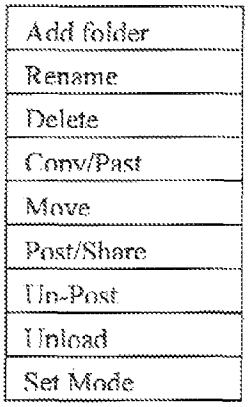


Fig. 9A

Fig. 9B an example of pop-up menu for file

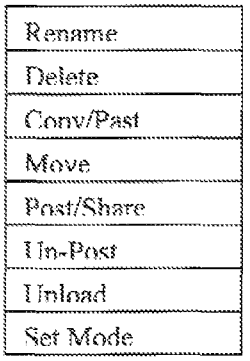


Fig. 9B

Fig. 10A an example of non pop-up operation menu for folder:

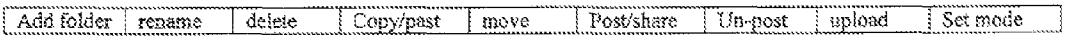


Fig. 10A

Fig. 10B an example of non pop-up operation menu for folder:

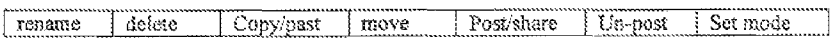


Fig. 10B

Fig. 11: an example of center view of list of posted files or folders in file & folder area of a private work space of a user X in a special personal share management section of a web-page.

File or Folder Name	On System	Share to Who	
My-video jpg file	System-1	Jane	Delete shared?
Project plan	Engineer-3	Jim	Delete shared?
C program code for web	Support-2	John	Delete shared?
Picture of building	System-1	Jane	Delete shared?
Engineer Drawing	Engineer-3	Jim	Delete shared?
Patent application-v1	System-1	Ted	Delete shared?
Folder 2	System 2 of system group-A	User Y	Delete shared?

Fig. 11

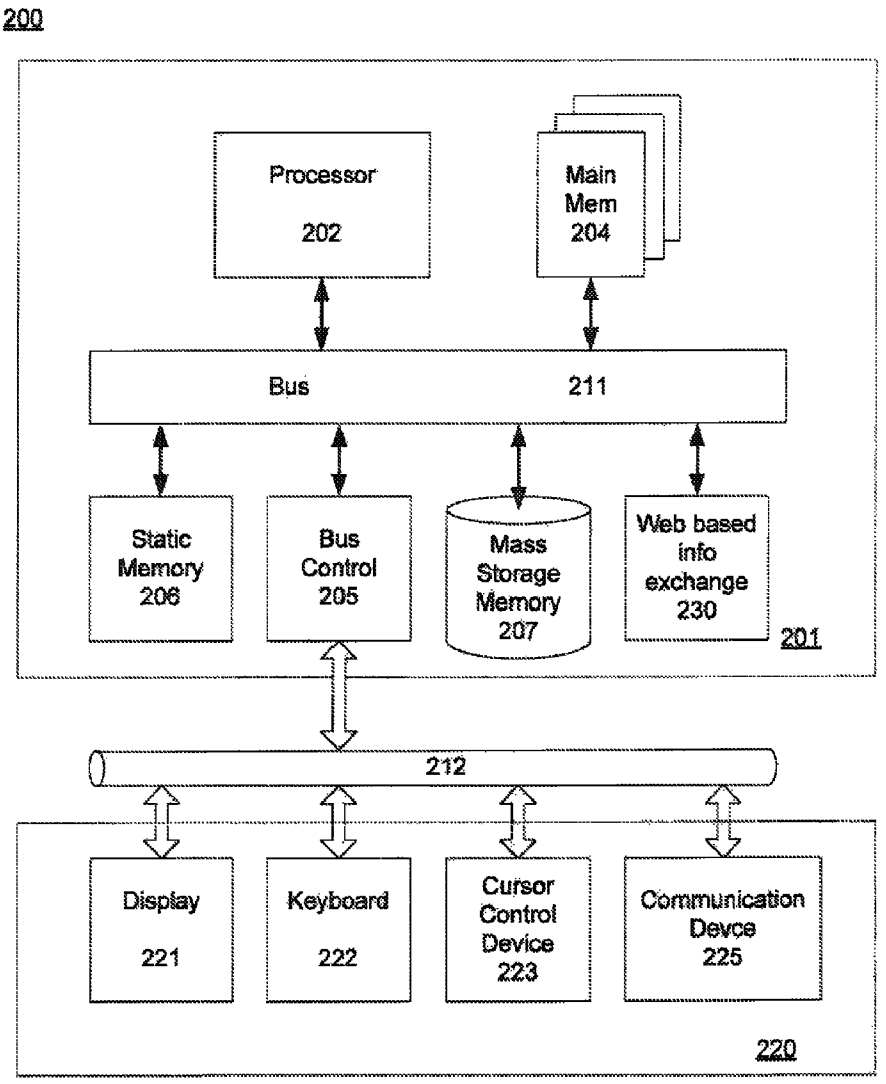

200
Processor
202
Main
Mem
204
Bus
211
Static
Memory
206
Bus
Control
205
Mass
Storage
Memory
207
Web based
info
exchange
230
201
212
Display
221
Keyboard
222
Cursor
Control
Device
223
Communication
Devce
225
220

FIG 12

METHOD AND APPARATUS OF DYNAMIC UPDATING WEB PORTALS

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation under 35 U.S.C. §120 of U.S. patent application Ser. No. 12/511,039, filed Jul. 28, 2009, which in turn is a continuation-in-part of (a) U.S. patent application Ser. No. 11/732,496, filed Apr. 2, 2007, which in turn claims the benefit of U.S. Provisional Application No. 60/787,653, filed Mar. 31, 2006, and is a continuation-in-part of U.S. patent application Ser. No. 10/713,904, filed Aug. 6, 2002, now U.S. Pat. No. 7,418,702, and U.S. patent application Ser. No. 10/713,905, filed Aug. 12, 2002, now U.S. Pat. No. 7,379,990, and (b) U.S. patent application Ser. No. 11/374,302, filed Jul. 2, 2004, now U.S. Pat. No. 7,945,652, which in turn is a continuation-in-part of U.S. patent application Ser. No. 10/713,904, filed Aug. 6, 2002, now U.S. Pat. No. 7,418,702, and U.S. patent application Ser. No. 10/713,905, filed Aug. 12, 2002, now U.S. Pat. No. 7,379,990, the entire disclosures of which are all hereby incorporated by reference in their entireties for all purposes.

FIELD OF THE INVENTION

The present invention generally relates to communications network. More specifically, the present invention relates to web based communications system.

BACKGROUND OF THE INVENTION

With increasing popularity of using the Internet and World Wide Web ("the Web") for the rapidly changing digital world, individuals as well as enterprises, exchange (or swap) large volumes of information through the Web. The demand for larger and faster information exchange has increasingly grown in recent years. For example, various business meetings and conferences are conducted over the Web with attendees scattered around the world. To enhance the communication between the meeting attendees, not only voice (and/or image) information is important, but also information exchange can be critical.

Fast, volume, and secure information exchange is important in enhancing the efficiency of a communications network. At Internet era, the web portals become a means for personal, group, or organizational communication. Therefore, an efficiency and cost effective method of creating and updating web portal becomes critical to each business and individuals.

Accordingly, there is a need in the art to provide a faster, more secure, volume information exchange over the Web.

SUMMARY OF THE INVENTION

The present invention relates to a method and apparatus for information exchange over a web environment.

With the development of central controlled distributed scalable virtual machine (CCDSVM) and the web-based computer user working (operating) environment (WCUWE), the problems mentioned in the previous section can easily be solved by introducing a "dynamic workspace" technology of this invention within the frameworks of CCDSVM and WCUWE.

With dynamic workspace technology, one or more login users of CCDSVM, each using a browser of any system and from anywhere, can instantly perform tasks. For example,

each user can post or un-post messages, files, folders, or other resources information on systems of CCDSVM with group-based model or non-group 1-to-1 model to one or more audiences on the network. This technology will also achieve larger scales of secure information exchange without size limitation and the installation of special software. Specially, this technology can dramatically improve the efficient and cost effectiveness for updating the contents of any existing web portals.

The implementation of this invention has provided users with a web-browser based system that is capable of securing exchanges of messages, files, folders, or other resources information including storage and network resources on systems crossing network domains without size limitation.

Additional features and benefits of the present invention will become apparent from the detailed description, figures, and claims set forth below.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be understood more fully from the detailed description given below and from the accompanying drawings of various embodiments of the invention, which, however, should not be taken to limit the invention to the specific embodiments, but are for explanation and understanding only.

FIG. **1** is a block diagram illustrating a layout of a CCDSVM with a central control management system and multiple provisioned systems, client system and console system in accordance with one embodiment of the present invention.

FIG. **2**A is a block diagram illustrating a layout of a CCDSVM with which users from their web-browser of provisioned system can communicate with each other in accordance with another embodiment of the present invention.

FIG. **2**B is a block diagram illustrating a layout of a CCSDVM with only a single control management system to be accessed by massive client systems in accordance with another embodiment of the present invention.

FIG. **3** is a block diagram illustrating an example of a software for a CCDSVM in accordance with one embodiment of the present invention.

FIG. **4**A illustrates an example of a dynamic workspace in accordance with one embodiment of the present invention.

FIG. **4**B illustrates an example of possible resources which may be assigned to a user-group common workspace in accordance with one embodiment of the present invention.

FIG. **4**C illustrates an example of a user private workspace and possible assigned system resources in accordance with one embodiment of the present invention. The mentioned managing resources can be realized via operation menu associated with displayed resource information. For example, a file-folder can be associated with an operation menu containing an operating option of creating a file, which encoded into a web page and to be displayed in user's web browser on his/her native system. Upon the user to select and submit the task of creating a file, an editor can be invoked and displayed in user's web browser or in a separate displayed area, so that user can input the content of the new file and further save it via a save new file operating option to transmit the created file back to his/her default home file-folder/directory.

FIG. **5** illustrates an example of resource objects on systems of CCDSVM in accordance with one embodiment of the present invention. The resource information can be network resource such as the identification information of one or more provisioned systems, which controlled and grouped by the control system if said control system is not an standalone

system; wherein said identification information of provisioned system includes system name, IP address, ID such as system service group ID.

The resource information also can be the information of the hardware resources on the provisioned system and control system such as one or more network devices such as Ethernet or wireless cards, storage devices such as SCSI or RAID disk, memories, CPUs, and one or more user interface (input/output) devices such as keyboard, mouse, monitor and displaying devices, and audio and video recoding or playback devices and components including digital camera or video and audio encoding cards etc. and there is no limits.

The resource also could be data resource on said system such as one or more file systems built on one or more storage devices, one or more file-folders and various type of data files or application files on said each file system, or one or more record file such as user account or user-group profiles or one or more message data, wherein said data file can be document file such as Word or PDF or Spreadsheet or PowerPoint or plain text file, or media data of streaming video or audio file including AVI, MPEG, MP3 or JPEG image file, and said application file is binary file such as for web application.

The resources can be kept into one or more information lists and to be stored in database on the storage media of the control system. The database is organized as one or more lists of tables, wherein each of the lists contains a certain type of resources or mixed type of resources. The storage media could be a system's internal storage such as memory or disk drive, for example.

The resource information of this invention can be encoded, displayed and operated as web folder-tree in web browser using multi-layered item list (MLIL) in the memory of control system, wherein each node of said MLIL binds with corresponding operation menu, attributes of said resource and graphic image, wherein said attributes of said resource are resource's name, ID, size, location or address, time stamp, ownership, and so forth without limit. Each node of the MLIL can have its next layer of one or more sub-nodes representing another layer of one or more actual resources.

The resource MLIL can be encoded into web page and be displayed in user's web browser such that said user can interactively select each permitted resource node of MLIL and to perform desired tasks through said operation menu. The list of resources displayed as web folder-tree has been demonstrated in embodiments of present invention such as depicted in FIGS. **7**A, **7**B, **8**A, **8**B, and **8**C. The folder-tree can be organized and displayed including one or more resource nodes, which are organized into one or more layers and each layer is vertically aligned, right indented and displayed below the layer above it, wherein each node at each layer represents a corresponding resource and can have its next layer one or more nodes for representing the corresponding resource's next layer one or more resources if there are any.

FIG. **6**A illustrates an example of displaying a web-page with 4 sections in web-browser of user X in user-group-1 during an interactive online meeting in accordance with one embodiment of the present invention; Each section of a web page can be updated and refreshed automatically and be updated/refreshed independent of other sections by a runtime control script such as a Java script or others, which can be preloaded into the web browser via an encoded web page. Of course, the entire web page also can be updated and refreshed manually by user or by a different control script.

FIG. **6**B illustrates an example of displaying a web-page with 4 sections in web-browser of user Y in user-group-1 during interactive online meeting in accordance with another embodiment of the present invention. As mater of the fact, it has illustrates that each web page can be divided into multiple screen sections and each section can be encoded with resources information actually from one or more user's private workspace.

In one embodiment as illustrated in this figure, the user-group-I section encoded with multiple files in group common workspace, which actually posted from three private workspaces by three corresponding users in the group. Also, only the actual owner of each file can from own private workspace to withdraw his previously posted file. For example, the picture I was posted by user Y from his private workspace, therefore, only user Y from his private workspace can be permitted to un-post this previously posted picture I. Similarly, the user-group-I section being encoded and displayed with multiple messages in group common workspace, which actually posted by multiple users from each of own multiple private workspace. Besides, each message can be un-posted by the actual owner of the message from the owner's private workspace.

FIG. **6**C illustrates an example of displaying a web-page with 4 sections in web-browser of user X in user group-1 before interactive online meeting in accordance with another embodiment of the present invention.

FIG. **6**D illustrates an example of displaying a web-page with 4 sections in web-browser of user Y in user group-1 before interactive online meeting in accordance with another embodiment of the present invention.

FIG. **6**E illustrates an example of a displayed public web-page that each massive online user can access and view from each of their browser without login to CCDSVM during said user X and user Y in user-group-1 online meeting in accordance with another embodiment of the present invention.

FIG. **7**A illustrates one-to-one, peer-to-peer post or un-post messages and folders of user X in accordance with one embodiment of the present invention. As matter of the fact, this also has illustrated that a section of web page can be encoded with information in multiple private workspaces because the displayed user-X's private workspace also includes information posted from another user-Y's private workspace such as "Sales folder", which only user-Y can withdraw it via his private workspace.

FIG. **7**B illustrates one-to-one, peer-to-peer post or un-post messages and information of folders of user Y in accordance with another embodiment of the present invention.

FIG. **8**A is an example of a web-based graphic presentation view of assigned resources in a user X's private workspace in accordance with one embodiment of the present invention.

FIG. **8**B is an example of a web-based graphic presentation of assigned resource in the private workspace of user Y in accordance with another embodiment of the present invention.

FIG. **8**C is an example of a web-based graphic presentation view of user Y's private workspace after user X dynamically posted and shared "folder **2**" in user X's private space in accordance with another embodiment of the present invention.

The FIGS. **7**A, **7**B, **8**A, **8**B, and **8**C also have illustrated that how does the web folder-tree can be used to organize resources in workspaces for efficient operating and managing resource during communication and sharing activities. The web folder-tree can be operated in similar way as native window based folder-tree; A user can select a node to expand and display its sub-nodes if there are any, and said user also can select a previously expanded node to collapse and hide all its sub-nodes from displaying. Therefore, said user can continuously browser and expand said sub-node of a folder-tree as long as there are next level of resource nodes until find a

targeted resource node. Since each node is bound with operation menu associated with one or more options of different tasks, the entire displayed resource folder-tree can be conveniently accessed and operated.

FIG. **9**A is an example of a pop-up menu for folder in accordance with one embodiment of the present invention.

FIG. **9**B is an example of a pop-up menu for file in accordance with another embodiment of the present invention.

FIG. **10**A is an example of a non pop-up operation menu for folder in accordance with one embodiment of the present invention.

FIG. **10**B is an example of a non pop-up operation menu for folder in accordance with another embodiment of the present invention.

FIG. **11** is an example of a center view of a list of posted files or folders in sharable file & folder area of a common workspace of a user X, through witch the user X can smoothly manage the massive sharing information for participating one or more 1-to-1 communication with different communication partners, or participating group model communication in accordance with one embodiment of the present invention.

FIG. **12** has illustrated a computer system and its components.

DETAILED DESCRIPTION

Embodiments of the present invention are described herein in the context of a method, system and apparatus for providing a web based communications network with fast, volume, and secure information exchange. Those of ordinary skill in the art will realize that the following detailed description of the present invention is illustrative only and is not intended to be in any way limiting. Other embodiments of the present invention will readily suggest themselves to such skilled persons having the benefit of this disclosure. Reference will now be made in detail to implementations of the present invention as illustrated in the accompanying drawings. The same reference indicators will be used throughout the drawings and the following detailed description to refer to the same or like parts.

In the interest of clarity, not all of the routine features of the implementations described herein are shown and described. It will, of course, be appreciated that in the development of any such actual implementation, numerous implementation-specific decisions must be made in order to achieve the developer's specific goals, such as compliance with application and business-related constraints, and that these specific goals will vary from one implementation to another and from one developer to another. Moreover, it will be appreciated that such a development effort might be complex and time-consuming, but would nevertheless be a routine undertaking of engineering for those of ordinary skill in the art having the benefit of this disclosure.

In accordance with the present invention, the components, process steps, and/or data structures described herein may be implemented using various combined types of operating systems, computing platforms, computer programs including various software modules, and/or general purpose machines. In addition, those of ordinary skill in the art will recognize that devices of a less general purpose nature, such as hardwired devices, field programmable gate arrays (FPGAs), application specific integrated circuits (ASICs), or the like, may also be used without departing from the scope and spirit of the inventive concepts disclosed herein. Where a method comprising a series of process steps is implemented by a computer or a machine and those process steps can be stored as a series of instructions readable by the machine, they may be stored on a tangible medium such as a computer memory device (e.g., ROM (Read Only Memory), PROM (Programmable Read Only Memory), EEPROM (Electrically Eraseable Programmable Read Only Memory), FLASH Memory, Jump Drive, and the like), magnetic storage medium (e.g., tape, magnetic disk drive, and the like), optical storage medium (e.g., CD-ROM, DVD-ROM, paper card and paper tape, and the like) and other known types of program memory.

Those of ordinary skill in the art will realize that the following detailed description of the present invention is illustrative only and is not intended to be in any way limiting. Other embodiments of the present invention will readily suggest themselves to such skilled persons having the benefit of this disclosure. It will be apparent to one skilled in the art that these specific details may not be required to practice to present invention. In other instances, well-known circuits and devices are shown in block diagram form to avoid obscuring the present invention. In the following description of the embodiments, substantially the same parts are denoted by the same reference numerals.

While particular embodiments of the present invention have been shown and described, it will now be apparent to those skilled in the art having the benefit of this disclosure that many more modifications than mentioned above are possible without departing from the inventive concepts disclosed herein. Therefore, the appended claims are intended to encompass within their scope all such modifications as are within the spirit and scope of this invention.

The present invention can be implemented by a web based system operated in a network infrastructure, such as a central controlled distributed scalable virtual machine ("CCDSVM"). FIG. **1** illustrates a CCDSVM in accordance with an embodiment of the present invention. A CCDSVM includes one or more groups of provisioned systems **3** that automatically forms one or more service groups/pools, a control management system **2**, console systems **1**, and client systems **10**. Control management system ("control system") **2** is configured to control one or more groups of systems ("provisioned system") **3**. Console systems ("console systems") **2** provide management while client systems **10** ("client systems") initiate accessing provisioned systems **3** and control system **2** via a web-browser over a network infrastructure, such as the Intranet, an Internet, and/or a LAN.

The network infrastructure ("network") includes all necessary hardware and software resources that facilitate various users from different geographic locations to communicate through the Internet, an Intranet, and/or a LAN. The hardware of the network includes network connection media, wireless media, circuitry components, and communication equipments. The network connection media includes various cables such as Ethernet and/or optical fiber. The wireless media is capable of establishing wireless communication links through air. The circuitry components include processor, data bus, memory devices, and/or circuit board. The communication equipments, such as switches, routers, gateways, and/or adapters, are used to provide a communications network. It should be noted that there are other possible elements of communication equipment that may be needed to form a communication link, but they are not necessary to understand the present invention.

The network of the CCDSVM provides the scalable bandwidth and computational powers for CCDSVM. In one embodiment, one or more switches and routers can be added and removed for the network of the CCDSVM for dynamically connecting and adding removing one or more provisioned system units for the service pools/groups of the

CCDSVM in response to the service and bandwidth and capacity requirements in providing services to meet the demand from more users on the client systems.

The software infrastructure includes Internet Protocol ("IP") addresses and system name identification software such as Domain Name System ("DNS"), firewall software, IP gateway set-up software, IP broadcast, and so forth without limit. The communication protocols over the network could be IP-based standard or non-standard proprietary protocols such as Hypertext Transfer Protocol ("HTTP/HTTPS") over Transmission Control Protocol/Internet Protocol ("TCP/IP") or Simple Object Access Protocol ("SOAP") over TCP/IP. It should be further noted that some IP based protocols are proprietary protocols and some are non-IP based protocols.

A web-browser is an application program that allows a user to access information over the Web, such as Windows Internet Explorer ("IE"), FireFox, Netscape, and/or Mozillar. A web-browser could also be any proprietary software, which uses web-protocols such as HTTP, WAP, SOAP for communication with web server over the network. The web-browser can be implemented with any or a combination of suitable programming languages such as C, C++, Java, or XML. In another embodiment, the web browser can be replaced with another networked user interface tool/application, which can be implemented using non web based specialized protocols.

A web page, on the other hand, is a displayable for information resource that is accessible through web-browser. In present invention, the web pages are being used for display the user interface part of the WCUWE for users from each of own web browser performing interactive tasks. The web pages may be hosted by a web server via a web server system and can be transmitted (or retrieved) by a web-browser. The web-page can be implemented with any or a combination of suitable programming languages such as C, C++, Java, or HTML, XML, WML and so forth without limit.

Provisioned systems **3**, client systems **10**, console systems **1**, and control management system **2** are computational systems, which could be server system, such as database server, web server, email server, video server, file server, storage server such as NAS or SAN or security monitoring devices with video and audio recording and playback devices, or desktop or laptop system, handheld wireless devices such as PDA, wireless phone, and so forth without limit. The computational systems include hardware resource of electronic components or devices such as CPUs, storage devices, input/output and display components and memory controller. The storage devices include internal memory such as read-only memory ("ROM") and/or random access memory ("RAM"). The storage devices may further include external storage devices such as magnetic disks or tapes, which include redundant array of inexpensive disk drives ("RAID"), just bunch of disk drive ("JBOD") and memory stick. The associated storage controller can be Integrated Drive Electronics ("IDE"), Small Computer System Interface ("SCSI"), Fiber optical controller, or a combination of above-mentioned devices without limitation.

The computational systems also include non-embedded or real-time embedded OS running, which could be Linux, or Windows, Unix, or proprietary OS. It should be noted that the present invention may include multiple control systems at different layers of system configuration topology wherein each of the control systems is capable of communicating with each other through a network. The control systems at middle layers play dual roles as both control system and provisioned system in the system configuration topology.

FIG. **1** illustrates a CCDSVM platform with multiple clients in accordance with one embodiment of the present invention. The CCDSVM includes console systems **1**, control management system **2**, provisioned systems **3**, and networks **11-12**, wherein networks **11-12** are used inter-connecting with all systems of CCDSVM and client systems **10**.

Console system **1** is a computational system having a web-browser or web-console **9** wherein web-browser is used by privileged users for accessing the CCDSVM platform. The web browser **9**, which could be on any system, permits a user to access information from the CCDSVM by following a web URL link In one embodiment. For example, a privileged user enters a URL of "https://69.107.28.123/stt/sttwebos" on a command line of web browser **9** and once the link is established, the user can obtain information hosted by the web-based computer user work/operation environment ("WCUWE"). Console system **1**, in one embodiment, further includes software modules **13**, wherein modules **13** may be used to facilitate communication between console system **1** and control management system **2**.

To support non web based networked user interface, the console system **1** can have a specialized software modules **13** using protocols other than mentioned web protocols. In another embodiment, if the control system **2** support a non-web based network computer user work environment ("NCUWE"), a user from a program on user's native system, which can be provided by the other software modules **13** of the console system **1**, can send a similar request to the control system **2** to indicate accessing a desired networked program of the NCUWE on the control system **2**.

Control system **2**, in one embodiment, is a server system, a desktop or a laptop system or even a wireless devices with respective memory and storage capabilities. Control system **2** includes web server software **7** and console supporting software **6**. Console supporting software **6** includes web server interface software modules **5** and control management software modules **4**. The control management system **2** may also have web-browser **9** used as a web-console **9** of control system **2**. Web server software **7** sends/receives data to/from web-console **9** of console system **1** or client system **10** or control system **2**. Web server software **7** is configured to provide secure sockets layer ("SSL") encryption to encrypting the data before transmission to enhance the security. Web server software **7** could be commercially available software such as Apache™ from open source, or IIS from Microsoft or a proprietary software. Web server software modules and console support software modules can be implemented by any suitable or a combination of any suitable programming languages such as C, C++, Java, JavaScript, HTML, or XML.

Console supporting software **6** communicates with service software modules **8** of provisioned systems **3**. Web server interface **5** of console support software **6** is capable of providing special functions that are otherwise performed by web server **7**. If no provisioned system **3** is coupled to the CCDSVM platform, control management system **2**, in one embodiment, is reconfigured to continue monitoring systems over the network while the CCDSVM platform is also reconfigured to a single stand-alone system, as shown in FIG. **2**B. Users can access and/or obtain objects or system resources information via web-browser **9**, which can be resided at any client system **10** or console system **1** or control management system **2**.

Provisioned system **3**, in one embodiment, is a computational system, wherein the system may contain suitable operating system ("OS"). Provisioned systems **3** can be automatically provisioned by control system **2** through a variation of the automatic system service pool construction protocol, which is a proprietary combined sequence of TCP/UDP/IP protocols as described in prior said application entitled

"Method and Apparatus for Web-based Storage On Demand" and to form one or more service pools (system groups) based on group ID. Provision process, also known as pooling process, is a method allowing control system **2** to control one or more network systems by reconfiguring the network systems. For example, when a provisioned system boots up via a communication protocol, control system **2** detects and obtains provisioned system's name, IP address, and system information wherein the system information includes network information, service group ID, storage information, file system information and so forth without limit, and further stores these information into a provisioned system unit information lists in a database on the control system **2**.

Each provisioned system **3** can be monitored, accessed, and/or operated by a user(s) through said web-browser **9** after the provisioning process. The control system **2** also updates status of provisioned system unit in the service pool in response to detect the shutdown or link down of any provisioned system and in response to admin staff's maintenance requests for removing any provisioned systems from service pools.

In an alternative embodiment, provisional system **3** is controlled by users through client systems **10** or console system **1** via control system **2** with proper users' authentication. Each provisioned system **3** contains service software modules **8**, which is used to communicate with control system **2**, console system **1** and client system **10**. For example, service software modules **8** of provisioned system **3** can communicate with control management software **4** to carry out tasks for monitoring, accessing or managing resources of provisioned system **3**. Service software **8** can communicate with web-browser **9** of client system **10** or console system **1** or control t system **2** to transfer data between them or deliver service to them, or to communicate with another provisioned system **3** to send or receive data.

In one embodiment, the CCDSVM provides the service software modules **8** to includes the functionalities of the console support software **6** and web server software **7** of the control system **2** to form the WCUWE of the provisioned system. Therefore, when a user login to the control system for access the CCDSVM, the user also can access the permitted WCUWE of the provisioned system **3** via a displayed entry point of the user's private workspace in the user's web browser. Further, each provisioned system **3** is able to provide data and application services for one or more client systems independent of other provisioned system **3** and without going through the control system **2** again.

In one embodiment, one or more spare provisioned systems **3** can be provided, where each spare system **3** is configured with identical service applications and data as one or more corresponding provisioned system **3** configured in each service pool, for providing fault handling. Once the control system **2** detecting a faulted provisioned system **3**, it can immediately instruct a corresponding spare provisioned system **3** to provide the service replacement for the faulted provisioned system **3** to provide continued service for one or more clients. In addition, using spare system has certain advantage, for example, one spare system **3** can serve one or more provisioned systems **3**.

Service software modules **8** may include special software modules having compatible functionalities of web-server software **7** of control system **2** and to dedicate handling HTTP/HTTPS protocol or other web protocols if there is needs for web-based communication with client **10** or with other provisioned system **3** or with control system **2**. In another embodiment, the service software modules can be implemented to communicate with non web browser based networked user interface by deploying specialized network protocols such as non-web based TCP/UDP/IP protocols. Software modules **8** could include commercially available web server software **7** or any proprietary software. The service software modules **8** could be implemented with any or combination of suitable programming languages such as C, C++, Java, or JavaScript.

Net **11** and Net **12** are network infrastructures of the CCDSVM that are capable of providing communication links between control management system **2**, console system **1**, client system **10**, and/or provisioned systems **3**. Net **11** and Net **12** comprises one or more switches and/or routers for providing scalable network, where the switches and routers can be added or removed along with adding or removing one or more provisioned systems in response to the services requirements. Client systems **10** may not be a part of CCDSVM, but with the permission and authorization, one or more users from web-browser **9** of client system **10** can login to CCDSVM and access the permitted resources of CCDSVM using web-browser **9** by following a web URL link of CCDSVM. To support non web based network user interface, the client system **10** can have a specialized software same as the software modules **13** of the console system **1**. While privileged users can obtain, manage, access, and/or operate system resources from CCDSVM through WCUWE, regular (non-privileged) users at client systems **10** may be permitted to access only limited system resources of CCDSVM. The prior applications entitled "Concurrent Web Based Multi-Task Support for Control Management System," and entitled "Display multi-layers list item in web browser with supporting of concurrent multi-users" together have described how to setup user security profile for one or more users interactively selects and accesses permitted resources and performs various permitted tasks via one or more displayed permitted folder-trees/MLIL.

CCDSVM configuration, in one embodiment, includes four data flow paths. The first data path is a communication link between web browser **9** of client host **10** (or console host **1** or control management system **2**) and web handling software on control system **2** (such as web-server **7** and console support software **6**). With this path of data flow, whenever a user sends a request from web-browser **9** to web-server **7** and further to console support software **6**, console support software **6** collects required information from target systems and converts (encode) them into standard structured information (web-format/web-page) for web communication by using any or a combination of suitable programming language such as C, C++, Java, Javascript, HTML, XML, WML and so forth without limit. The targeted system could be one of provisioned systems **3** or control system **2**.

The information collected by the console support software **6** of control system **2** could be, in one embodiment, the system status, storage information, network information, user authentication profile, the file system information, files & folders information on control system **2**, or any targeted provisioned system **3**. Console support software **6** then passes this converted structured information (web page) to web server software **7** and further transmits to web-browser **9** through communication link Net **11** and Net **12** so that it can be displayed and viewed by a conventional web browser **9**, as shown in FIG. **1**.

The communication protocol used between the web-browser **9** of client system **10** (or console system **1** or control system **2**) and the web server **7** of control system **2** could be HTTP, HTTPS (SSL encrypted HTTP protocol) or any suitable web protocols for web communication, which could successfully transmit the data on the web or could be other

standard or proprietary IP-based on non-IP-based protocols. This data path may be simply referred to as the console support software **6** transferring data to/from said web-browser **9**.

In one embodiment, if the control system **2** is configured with multiple network interface equipments, the console support software **6** detects from which network interface the requested data stream submitted by web browser **9** is coming from, therefore, it will provide the response web page back to the web browser **9** via the same communication link that includes the identified network interface equipment.

The second data path is the data flow through communication link between control system **2** and provisioned systems **3**. Requests targeted to a provisioned system **3** are passed from the console support software **6** of the control system **2** to the service modules **8** of the provisioned systems **3** through the communication link network **12**. When the requests are received by a provisioned system **3**, the service modules **8** of provisioned system **3** carry out the requested tasks, and if there is needs, the service software **8** of provisioned system **3** then send the response back to console support software **6** of control system **2**, and further let control system to provide response back to the user's request. In another embodiment, the service software modules **8** of the provisioned system **3** provide response back to the user directly without going through the control system again.

The implementation of the actual products of this invention on the second data path may use proprietary TCP/UDP/IP based protocols for the communication between provisioned systems **3** and control system **2**. However, other standards or proprietary IP-based or any suitable non-IP-based protocol are also possible. The communication protocols used between console support software **6** of the control system **2** and service modules **8** of provisioned system **3** can be any suitable IP based or non-IP based protocols, whichever is suitable to transmit data between them. The typical data flow through this path could be the boot message, system status, network information, or storage information of provisioned system **3**, and so forth without limit as shown in FIG. **1**.

The third data path is the data flow between provisioned systems **3** and client system **10** or console system **1** via communication link **11**-**12**. Referring back to FIG. **1**, in one embodiment, the console support software **6** of the control system **2** may present a web link, which points to an object on a provisioned system **3**, to be displayed in web-browser **9** of client system **10** or console system **1** or control system **2**. The object pointed by the web link on the provisioned system **3** could be a text file, MPEG video, digital picture, PDF document, MS Power Point, Word documentation, etc. without limit. It also could be a link of another web service program. From web browser **9**, a user can directly access the information on a provisioned system **3** pointed by the web link without go through the control system **2** again. In this case, service modules **8** of provisioned system **3** also include web server software modules or equivalent one to directly support web browser **9** as mentioned before. The communication protocols with the third path could be web protocols such as HTTP or WAP. Alternatively, the communication protocols can be other standards or proprietary IP-based or non-IP-based protocols. To simplify the discussion, the data transmitted on this path will be mentioned as service software module **8** that send data to or receive data from said web-browser **9** or vice versa.

The fourth data path is data flow through communication link between provisioned system **3** and another provisioned system **3**. With this path, service modules **8** of one provisioned system **3** can directly transmit data or information to service modules **8** of another provisioned system **3** via a communication link Net **12** without going through control system **2**. The implementation of the actual products of this invention, for example, uses a proprietary IP-based protocol for communications between the two provisioned systems **3**.

The principle of this invention, however, does not rely on or limit to proprietary protocol. It should be noted that other standard or proprietary IP-based or any suitable non-IP-based protocol may also be possible. The data and information transmitted through the fourth data path can be various types of data. For example, a user on web-browser **9** may open a file folder on a provisioned system **3**. Further, the user can also instruct to select and transfer any file or file-folder from a current target provisioned system **3** to another provisioned system **3** by an interactive click mechanism. The data file or file-folder will then be transferred directly between the two provisioned systems **3** without going through control system **2**.

FIG. **2**A illustrates a variation of CCDSVM platform for web browser-based communication in accordance with one embodiment of the present invention. CCDSVM platform illustrated in FIG. **2**A is similar to the platform illustrated in FIG. **1** except every provisioned systems **3** includes a web-browser **9**. Also, the client system **10** can have same software modules **13** as console system **1** has. Thus, with proper authentication process, each user of provisioned systems **3** may access, manage their own provisioned system **3** from web-browser **9** either on their own local provisioned system **3** or on any client system anywhere on the network. It should be noted that when a user is registered with a provisioned system **3**, the user is authorized to access provisioned system **3** locally but not with CCDSVM. On the other hand, when a user is registered with CCDSVM, the user is not only can access its own local provisioned system **3** but also can be authorized to access multiple provisioned systems **3** and control system **2** of CCSDVM.

FIG. **2**B illustrates another example of a variation of CCDSVM, in which there is no provisioned systems **3** in CCDSVM. Control system **2** can run on its own and CCDSVM has degenerated into a single stand-alone system (FIG. **2**B). The users through web-browser **9** on any client system **3** (or on console system **1**, or on stand-alone control system **2**) are allowed to access and manage the stand-alone control system **2** via the WCUWE of the standalone control system **2**.

Unless specifically specified, the console support software modules **6** of control system **2**, the provisioned systems **3**, the service software modules **8** of the provisioned system **3**, and the web-browser **9** of the control system **2**, client system **10**, and console system **1** are referred to the CCDSVM platform illustrated in FIG. **1**.

A web-based computer user work/operation environment ("WCUWE") of CCDSVM is provided by a software infrastructure, shown in FIG. **3**. The WCUWE provides each permitted user to log into CCDSVM and further provide each login user a private web-browser based operating environment, which is a private workspace including one or more assigned and permitted resources on the control system **2** or provisioned systems **3** on the network.

In one embodiment, the user private workspace can be initially assigned and displayed in web browser with limited resources such as an entry point of the user's top level home directory for managing one or more next levels of file-folders and their files, an entry point for update user's password or managing one or more user's authentication, an entry point for accessing networking, and entry point for monitoring and managing hardware devices including storage devices, an entry point for joining 1-to-1 personal communication and an

entry point for joining group collaboration including accessing on-the-fly streaming video provided from other user's digital camera or archived streaming videos for the meeting, and so forth without limits. The user private workspace can be encoded into web page and displayed into user's web browser upon user's login to the control system for accessing the CCDSVM with any suitable or a combination of suitable programming languages.

In another embodiment, the owner (assigned user) can dynamically managing resources in the private workspace such as create a new file under a file-folder by using editing or other proper application tools, delete one or more files or file folders, copy or move files or file-folders from a source file-folder either on the control system **2** or on any provisioned system **3** in CCDSVM into the user's permitted file-folders/directories of the CCDSVM, or upload files from user's native system's local drives into any user's permitted file-folder/directory, or transmit files or file-folders from user's native system's local drives into the user's default home file-folder/directory structure via a specialized file and file-folder transfer mechanism.

In one more embodiment, the user's private workspace in the CCDSVM displayed in web browser also can be connected to the user's local file-folder/storage drives on the user's native system via the communication between the console support software modules **6** and the service software modules **8** on the user's native system as depicted in FIG. **2***b*. Therefore, the file-folders on storage drivers of a user's native system can be displayed and be accessed via the user's private workspace of the CCDSVM.

WCUWE further provides each user from a single web-browser **9** to interactively perform various permitted tasks and operations over various system resources (FIG. **5**), wherein the tasks can be performed concurrently from a single web-browser **9**. The tasks include user authentication management such as allowing privileged users to create users, groups, assign users to each group, to assign initial resources to each user or group, and to save the information of created users and groups into a database stored on a storage media of the control system.

The information of users or groups include user names, user IDs, assigned security permissions, passwords, access points, group names, group IDs, maximum members in a group, and so forth. The assigned security permissions include the user's role and credential information, which further includes from where the user can access CCDSVM, what service group/pool the user can access, what system and what specific resources on a system the user can access, and what task the user can perform. The user's security permission is encoded into a security data structure, where one or more fields of the security data structure containing one or more bits with bit-status of on or off to represent the status of one or more specific credential or permission for the user.

The privileged user can dynamically grant or revoke other user's security permission and credential via interrogating each user's security profile by inspecting, checking, and changing the status of each security permission field, or assign and deny each specific user to access the resources of the CCDSVM via set access control option from operation menu associated with any resource.

In addition, the console support software **6** of the control system **2** keeps tracking records for each user's every login session including the time of login and the time of log out, at what time performed what task or accessed what data or applications. In one embodiment, the console support software modules **6** receives and parses each user's accessing request, and writes information of each user's request into a access record in the per user secure access file, which can be linked to each user's security profile for real-time fast interrogation. In one embodiment, the per user secure access record file can be backup by copying it to another file with a special name tag, and it can further be emptied on a periodic bases for keeping the up to date user's access records.

The tasks also include personal or group communication, system and network management, accessing and managing data, service, and hardware resources including network and storage device, file system, file-folders, files, messages, user security, and so forth without limits. Many of tasks has been described in said prior applications of "Concurrent Web Based Multi-Task Support for Control Management System," and "Display multi-layers item list in web browser with supporting of concurrent multi-users".

After receiving each task information transmitted from the web browser **9**, the console support software **6** of control system **2** parses and stores each task information into an available entry in an user space task list. After completing said task to be executed on targeted system, the corresponding entry stored with said task on the user space task list is cleared up. In addition, the console support software **6** of control system **2** immediately provides a response web page to each web browser in response to each submitted task independent of the completion of the task execution.

In one embodiment, said response updates the original web page in said web browser to be encoded with the status of the task including the task result if the task is successfully completed or include the task execution status if the task is failed or in progress or include a location that said task result will be deposited. Additionally, the WCUWE deploys a lock protecting mechanism for said user space task list and all resources related to each task's execution in the event that multiple tasks may concurrently access and change the state of each mentioned resource.

The WCUWE of the CCDSVM includes software modules of present invention. For example, the software includes console support modules **6** of control system **2**, which further includes web interface/client interface modules **5** for receiving requests from end users' web browser either on the control system or remote system, and includes distribution control management software modules **4** for control, monitor, and manage the provisioned system units **3**. The distributing control management modules also controls user authentication, fault detecting and recovering, contents distributing, and tasks distributing and executing.

The software also includes the service software modules **8** of provisioned system **3**, other service software **13** of console system **1**, which can be used for send or receive data to/from the control system **2** and provisioned system **3** using non web-based protocols without web browser's involvement. The current implementation of WCUWE is based on a proprietary design of this invention. However, the nature and spirit of this invention does not limit to proprietary designed software. The WCUWE of CCDSVM also includes other proprietary software modules or conventional software such as web-server **7** of control system **2**, web-browser **9** of the control system **2**, the client system **10**, and console system **1**.

The WCUWE of the CCDSVM can be implemented with any suitable or a combination of various suitable programming languages such C, C++, Java, JavaScripts, HTML, XML, and so on. To simplify the discussion, the WCUWE will be used to represent software modules in CCDSVM described above.

Users of CCDSVM may be created by CCDSVM though a given process or any users on the Internet, Intranet, LAN without registering with CCDSVM. The WCUWE of

CCDSVM provides user to access various resources and perform various tasks through a preferred model of clicking on graphic or text represented resource objects on browser.

WCUWE is designed to be operated on a CCDSVM platform, and specially is designed to provide dynamic workspace (FIG. **4**A) to each user associated with various CCDSVM as depictured in FIG. **1**, FIG. **2**A and FIG. **2**B. WCUWE further facilitates each user to performs tasks for instantly posting a resource information to or un-posting/withdrawing user's previously posted resources information for web page's contents in a group model or in 1-to-1 model for users of CCDSVM in a same group or without group membership, or for massive public online users over the network, wherein said public users from each of own web browser do not have accounts with the CCDSVM and can access the public resources provided by the CCDSVM via public web pages.

The dynamic workspace of WCUWE is a centrally controlled collection of each group's workspace and/or user's private workspace. Console support software **6** of control system **2** of CCDSVM will assign each user a private workspace and each group a common workspace when each user or group account is created by a privileged user after the WCUWE of the CCDSVM enters into an operational mode, wherein a first privileged user is created during installing said software of CCDSVM. When a user or a group account is created, each user is assigned to a specific user security credential and permission and permitted resources based on each user's role. For example, some users are privileged user who can perform system administration tasks and others are general users who can be permitted to access certain types of applications and resources of the CCDSVM.

A workspace is an organized information structure, which can be saved on storage media including memory or disk drives or both of memory and disk drives by console support software **6**. A workspace is created during a privileged user performing the tasks of creating user or group. The workspace will be initiated and assigned with limited resources of CCDSVM such as shown in FIG. **5**. Various resources on either provisioned system **3** and/or on the control system **2** include respective memory, system-groups, systems in system-group, file system, file-folder, files, storage media, network media and so on without limits. The common workspace **15**, shown in FIG. **4**A and FIG. **4**B, of each group may be initially assigned with resources information (FIG. **5**) including users in this group, message area, file systems, file-folders and files on a specific provisioned system **3** or on control system **2**, and so on. The resources information of the common workspace may also be dynamically assigned or allocated by permitted user after creating group account. For example, during an online interactive meeting, a user posts a file information to the common workspace, so that a new entry of a file node is added to a file and a folder tree area of common workspace.

In addition, each group may be assigned to one or multiple users, wherein each user can also be assigned with a private workspace **16** of FIG. **4**A and FIG. **4**C during user account creation time. Also, one or more users also can be dynamically removed from a specific group depends on the needs. The private workspace may also be initially assigned by resources information of the CCDSVM during user account creation time. Thereafter, the private workspace is dynamically assigned with the resources of CCDSVM after user login CCDSVM.

After creating user and group accounts information, the privileged user can dynamically grant or revoke the permission for a general user to access one or more mentioned resource from his/her private workspace. In one embodiment, the privileged user can select a node containing a list of storage device on a hardware resource folder-tree of the CCDSVM, and further to select posting option from operation menu and input a targeted user information for binding the list of storage device on a system of the CCDSVM to a user's account or select a withdrawing option to remove the list of storage devices from user account profile. Therefore, during the operation, at one second, the targeted user may see said list from user's web-browser, and at another second, a general user will no longer see the list, which was in the user's displayed private workspace in web browser. Also, if an authorized user can access a list of storage devices on a provisioned system and further can take one of devices offline so that the device list on the provisioned system will have one less devices to be displayed in the user's web browser.

In another example, at one second, a privileged user grants a general user to access a file J in folder N on system M of CCDSVM by selecting the file J and selecting an set access control option from the web operation menu to submit task of set access control for said user, and at another second, the privileged user with similar steps to resets previous permission, thereby, the general user can view and access the file J in folder N on system M of CCDSVM at one time, and a second later, the general user will not be able to view and access it. The actual implementation of console support software **6** of control system **2** has achieved this capability.

The total number of users and user groups can be created is dependent on the capacity of the control system such as how larger the memory and storage capacity are on the control system. For example, if a user private workspace requires 1 Gig-bytes in storage space and 2 Meg-bytes in memory space, a system with 1-GB memory and 500 GB storage roughly can support 500 users. Similarly, the number of the concurrent users access CCDSVM over the network with a certain data rate also is bound by the network bandwidth that CCDSVM infrastructure can provided.

When a user account is deleted, the private workspace and user' membership with a group are automatically removed and updated. On the other hand, when a user group is deleted the group common workspace is deleted, along with removing the group membership encoded in each associated user's security profile without effecting the rest of user's private workspace. In another embodiment, when a different type of user group is deleted, all accounts of the associated users and their each of the private workspace in the group are deleted together with the deleting of the group account.

In addition, one or more types of resources in the workspace can be organized as one or more list of tables and can be stored in storage media of the control system as database. The integrity of security of each group is protected by limiting different resources to different group. As such, each group is authorized to have a limited view over assigned system resources on provisioned systems **3** as well as control system **2**. The same security integrity at user level can be achieved by assigning different system resources of provisioned systems **3** and/or control system **2** to different individual user based on exclusively assigned security permission and credentials. Therefore, each user is authorized to access and view the assigned resources.

FIG. **4**B has shown an example of assigned resources information of a group common workspace **15** including users area **17**, message area **18**, and folders and files area **19**, and others **20** in common workspace **15** of a group. All resource information in the group common workspace can be shared by all users in a same user-group. All resources in each group's common workspace can be selectively displayed in web-

browser of each login user, who belongs to same user group after said resources information in said group being converted to presentable web-format (web-page) by the console support software **6** of control system **2** of CCDSVM.

FIG. **4**C has shown an example of resources in private workspace (**16**) of a user, which has assigned private message area **21**, private file, folder area **22**, and private network or storage or other information area **23**. The assigned resources only can be exclusively viewed and accessed by the designated user unless he/she agrees to share with peer user crossing group boundary or with peer user in same group. The directory is a data structure which can hold one more files and file-folders at logical file system level and be stored on physical storage media.

FIG. **8**A has shown an example of displaying an actually implemented portion of web-page encoded with assigned resources information folder-tree in private workspace of user X, which includes three layers. The top layer includes a system-group-A **33** on said network, which includes its next layer of multiple provisioned systems **3**. One of provisioned system **3** (system-**2**) includes its next layer of two private file-folders **32** and **34** of user X, and each file-folder contains its next layer of files displayed in the private workspace of user X.

FIG. **8**B has shown another example of displaying an actually implemented portion of web-page encoded with assigned resources information in private workspace of user Y, which includes control system **2**, private folder and files **35** of user Y in a private workspace of user Y.

The private workspace of user X and user Y may also include private message area **21** and other resources, which are not displayed in FIG. **8**A & FIG. **8**B because WCUWE is designed to dynamically display part of user's private workspace in response to each user's needs.

It is notable that said user X and user Y, may be in same user-group such as said group-1, however, each of them has displayed a significant different private workspace on the control system **2** due to they have been assigned with very different resources of the CCDSVM depending on each user's role and security permission, which setup by privileged user of CCDSVM through web-based user interface provided by said console support software **6**. This is the beauty of the invention of WCUWE for CCDSVM.

The FIG. **6**C and FIG. **6**D are examples of web-page view in said web-browser **9** of user X and user Y in user-group-1 before an interactive online meeting. As shown in FIG. **6**C and FIG. **6**D, user X or user Y or other users (who's web-page/browser did not depicted) in same user-group-1 have entered a group online meeting after their login to CCDSVM from each browser on each of own native system over said network. Both users X and Y get an identical web-page provided by said console support software **6** of control system **2** except with different contents in displayed private workspace section of web-page due to each user has assigned with different resources. The web-page screen layout provided by said console support software **6** contains four sections:

1) a private user message input area (section-1),
2) a message displaying section for common message area **18** of FIG. **4**B for all users in user-group-1 (section-2),
3) a file or folder display section for group common file and folder area **19** of FIG. **4**B for all users in same user-group-1 (section-3), and
4) a file and folder tree displaying and management section (section-4) for user folder and file tree in private file and folder area **22** of FIG. **4**C of a user's private workspace.

Initially, said section-1, section-2, and section-3 are all empty due to no one has post any file information or message yet. The console support software **6** gets each login user's portion of assigned folders and files for online meeting from stored private file and file-folder area **22** of FIG. **4**C in private workspace of each user on said storage media, and further convert/encode said file or folder information into said section-4 of presentable web-format (web-page) to be displayable in web-browser **9** of each user. The choice of total four sections of web-page layout and the order of each section on the browser screen is irrelevant to this invention since other choices of design are also possible, and further, this invention does not have limitation on what web-screen layout shall be. The information of file or folder attributes includes the system name where the file or folder resided, the named and path of the file or folder, the owner of file or folder, the time stamps, the size of the file and so on without limitation.

During the interactive online meeting, said console support software **6** provides each user to post or un-post files or folders information to said section-3 of displayed group's common documents and contents management section with several steps. First, for example, providing said user X to click on a targeted file "Doc **1**" **24** or folder one a time in section-4 of displayed said file and folder management area of said user private workspace. Second, select the operation of either post or un-post from an operation menu (see FIG. **9**A, FIG. **9**B, FIG. **10**A and FIG. **10**B). Finally click on the selected operation to submit the task of either post or un-post on selected file or folder.

Each user with permission may perform un-post operation if such user has at least a previously posted file or folder information, which has already displayed in section-3 of common document and content management section of said web-browser **9** of all login users in said same user-group, see FIG. **6**A or FIG. **6**B for example.

If a file is selected, said file operation menu is used and if a folder is selected, the folder operation menu is used respectively. It is also possible to only provide one menu for both file and folder operation by said console support software **6**. The operation menu either can be pop-up menu for folder as shown in FIG. **9**A or pop-up menu for file as shown in FIG. **9**B, or non-pop-up menu for folder as shown in FIG. **10**A or non-pop-up menu for file as shown in FIG. **10**B, or drop-down menu or just one or more submit buttons. Each operation menu contains one or more operating options for user to select and submit corresponding task. The choice of operation menu is irrelevant to this invention although the actual implementation of this part of invention has used pop-up menu, and as matter of the facts that there is no limitation on how the operation menu is to be designed in this invention.

In the example of FIG. **6**A, user X can post a file named "Doc **1**" **24** to said section-3 of common document and contents display section of said web-browser **9**, In the example of FIG. **6**B, user Y also can post a file named "picture **1**" **25** to common document and contents display section of said web-browser **9**. In addition, another not depicted user Z posted a file of "project code". The files or file-folders could be one on said control system **2** or on provisioned any provisioned system **3** of CCDSVM.

In addition, during the interactive online meeting, the console support software **6** of control system **2** also provides each user to post message to one or more users in same group from said section-1 of private user message input section after typing a message and click on the "send" button. In the example of FIG. **6**A, user X posted a message of "Msg**1** of user X" **26**. In the example of FIG. **6**B, user Y also posted a message of "Msg**2** of user Y" **27**. In addition, another non depicted user posted a message of "Msg**3** of user Z". As a

result, all of three messages were displayed in the same said section-3 of each user's browser.

In a specific situation, the console support software **6** also provides permitted user to un-post the previously posted message. Just for example, in FIG. **6**A, said console support software **6** provides permitted user X to select a previously posted "Msg**2** of user Y" **27** from said section-2 of common message display section of said web-browser **9** of user X, and select un-post operation from said operation menu and submit the selected un-post task.

The detailed information of how the post or un-post task to be implemented will be discussed in a few paragraphs later. In addition, with this invention, the action of the "post" information such as posting information of files or folders or messages or other resources is equivalent to the action of "share" information. The action of "un-post" information is equivalent to the action of "un-share" previously posted or shared information, in other words, "remove" previously posted or shared information.

After user transmitting information data of the post or un-post task from web-browser **9** of console system **1** or client system **10** or control system **2** to control system **2**, the console support software **6** of control system **2** obtains and parses the data to determine: who is the original user initiating the task; what type of task; what type of associated resources for task to work on; who is the targeted user; and what is the time stamp, and may also include other information without limitation. The console support software **6** further determines that if the task is to post a file or folder from said original user, the console support software **6** picks up the corresponding file or folder information from file and folder area **22** of FIG. **4**C of private workspace of the original user and deposits such information to the file and folder area **19** of FIG. **4**B of common workspace of said group in this case.

If the task is to un-post/withdrawing a previously posted file or folder, which posted from a private workspace of a specific user, in said common workspace of said group, the console support software **6** search for the corresponding information of file or folder in the file folder area of said common workspace of said group and further remove the entry of matched information from said common workspace of said group after it is found.

If the task is to post a message to users in said group, the parsed message will be stored in private message area **21** of FIG. **4**C of original user's private workspace **16** of FIGS. **4**A and **4**C and parsed message also is copied to message area **18** of FIG. **4**B of common workspace **15** of FIGS. **4**A and **4**B of said user-group by said console support software **6**.

If a task is to un-post (delete/remove) a previously posted message, the console support software **6** searches for a matching message in said common message area **18** of FIG. **4**B of said common workspace of said group and remove such message entry if it is found based on message and original user identification and other associated information such as time stamp, security permission and so on without limitation.

It shall be noticed that the task of un-posting a file or folder of this invention does not actually delete the original file and folder from original user's private workspace. In addition, a lock protection mechanism is deployed whenever a message or information of a file or folder is written, stored, or deposited to either said private space or said common workspace, or to said disk drives or other storage media. Throughout the rest of discuss, the lock protection is assume to be a default action without further mentioning whenever a write/store/deposit action take place. As matter the fact, each task of posting file, folder, and message as well as task of un-posting the previously posted file or folder, or message performed by multiple users in multiple groups with this invention can be executed by the console support software **6** instantly with memory speed.

FIG. **6**A and FIG. **6**B illustrate that after console support software **6** of control system **2** successfully executing the tasks for each user in each group, any user in a said user-group can get a fresh view of said section-2 of the common messages display area of user-group-1, and a fresh view of said section-3 of the common documents & contents display area of user-group-1 from each user's said web-browser **9** on client system **10** of FIG. **1** or on control system **2** of FIG. **1** or on console system **1** of FIG. **1**.

User can manually click on a refresh-button, which does not show on FIG. **6**A and FIG. **6**B, to refresh entire web page screen including corresponding part of screen of either said section-2 or section-3. Also the corresponding part of browser screen of said section-2 or section-3 can be automatically refreshed. As mater of the fact, optionally, the console support software **6** can provide a run-time script to said web browser via a web-page, which is loaded into web browser after each user login to CCDSVM and start group online meeting.

The run-time script will automatically and periodically perform tasks of refreshing said section-2 and section-3 of web page screen. As mater of the fact, the script can control which section to refresh or refresh all sections in a web page. The frequency of the periodic refreshing can be adjusted by said console support software **6** of CCDSVM depends on the needs of the meeting, which typically can range from 1 to 3 seconds and there is no limitation on this aspect. Also, with the preferred example, said run-time script provided by WCUWE is a JavaScript, however, other type of programming scripts or methods are also possible and there is no limitation on what kind of run-time program shall be used in this invention. In addition, the console support software **6** provides each user the choices of either automatic or manual refreshing their said web-browser **9**.

Upon refreshing said section-2 or section-3 of said web-browser **9** screen of each user, a request of updating such screen will be generated and sent from said web-browser **9**, to control system **2**. The console support software **6** obtains and parses each request to determine what user-group, and which part of said common workspace of said group need to be updated.

Based on information of each parsed request, the console support software **6** of control system **2** retrieves the updated resources information from said common workspace **15** of FIGS. **4**A and **4**B of said user-group, which includes recent posted files, folders, or messages from each user in said group, and also includes the updated list of files and file-folders or messages after un-post one or more files or folders, or messages by users in same group. Finally, console support software **6** of control system **2** converts/encodes retrieved information of resources and objects into presentable web-format (web-page).

If the request is to update the group messages, the converted information is sent to the section-2 of group common message display area in corresponding user's said web-page/web browser **9**. If the request is to update the group common files and folders display area, the converted information is sent to the section-3 of said web-browser **9**. Therefore, after refreshing said sections on said web-browser **9**, each user can have an identical view of all posted files and folders from section-3 of group common document display area on said browser **9**, and can have an identical view of all messages from section-2 of group common message display area on said web-browser **9**, which posted by all other users in a same group. Meanwhile said console support software **6** still allow

each user kept most of resources information in their own private workspace un-exposed to other users in same group during the online interactive meeting.

For example, User X at his/her said web-browser **9** can see "picture **1**" **25** of FIG. **6**A posted by user Y in group common file and folder display area but not other information in user Y's private workspace area, and plus the "Doc **1**" posted by user X himself. Also, user Y at his/her said web-browser **9** can see "Doc **1**" **24** of FIG. **6**B posted by user X in group common file and folder display area but not other information in user X's private file and folder display area, and plus the "picture **1**" posted by user Y himself.

A difference between a conventional Internet group communication model and the group based communication model described in the present invention is that console support software **6** of control system **2** provides each user to control post or un-post said resources information from user's own private workspace while the conventional Internet communication model are not capable of performing posting/un-posting task. Also, for example, un-posting a previous posted a file or folder information with WCUWE of CCDSVM does not delete the file or file-folder information nor does delete the corresponding physical file or file-folder in user's private workspace while the delete operation with said existing Internet group communication model will completely delete a uploaded file in a group. The dynamic workspace of this invention represents a true beauty of the combined security protection of information and flexibility and efficiency of online meeting provided by WCUWE of CCDSVM to each users and groups in online meeting.

In addition, with a preferred example, since posted information of each files are encoded by said console support software **6** with a web link and displayed in said section-3 of web-browser **9**, therefore, each user in a user-group can click on said link to download the files, which provided in the private file and folder area of the private workspace of their peer users in group, from a system where the file is located. For example, user X can download said "picture **1**", which may reside on either control system **2** or any provisioned system **3** from user Y and user Y can download said "Doc **1**", which also may reside on either control system **2** or provisioned system **3** from user X at same time. This achieves a true peer-to-peer exchanging files on permitted system from permitted users in said user-group. In addition, there is no size limitation on the file size for file to be downloaded, which is a gifted benefit from Internet technology. Again, the web link discussed in present invention just represents a type of reference point to an object for user to access, which can be accessed in a cross network environment. Therefore, the mentioned web link itself shall not be limited to the concept of the web only environment and it also can be referred to a way to reference an object in a non web-based networked environment.

Besides the exampled and understandable method of download, other methods of retrieve peer users' digital documents or contents are also possible by deploying a more efficient proprietary file transfer mechanism to reach the results of said exchanging file. It shall be noted that said method of download is only applicable to said files and does not apply to folder and other said resource information. To get a targeted peer user's entire folder and the underneath files and sub folders, a special files and folder transfer method is provided. Transferring files and file-folders across systems has been discussed in said prior application of "Display multi-layers list item in web browser with supporting of concurrent multi-users" via copy/paste mechanism.

As shown in FIGS. **9**A & **9**B, and FIGS. **10**A & **10**B, console support software **6** of control system **2** also provides each user to perform many other tasks during online meeting through said operation menu such as upload, rename, delete, copy/paste and so on without limitation. The upload operation allows each user to dynamically upload files one a time through web-browser **9** from local storage of client system **10** or console system **1** to any permitted file-folder in said user's private workspace on control system **2** or on provisioned system during meeting, and further to be posted to said common space of user-group each user belong to.

The user interactive click driven copy/past operation allows files and folders from permitted folders of permitted provisioned system **3** to be dynamically physically copied to each users' private workspace on control system **2**, and the information of said files and folders could further to be displayed in said web-page/browser of each user and to be posted to common workspace of group, which each user belong to in addition to direct posting these file or folders without physical copying over. As matter of the fact, operations provided by console support software **6** of control system **2** have added efficiency to posting or un-posting files and folders for each user depending on the real needs.

Additionally, said console support software **6** provides each user of each group capability of saving the posted messages of common message area **18** of said common workspace of said each group into a file in common file & folder area of said common workspace of each group. Further, the saved file can be viewed, downloaded by each user in each group with same method of posting. Besides whenever there is needs, said console support software modules **6** provide users to reset message area to be empty by removing all message entries in message area of said user-group's common workspace.

The important fact is that the designed WCUWE of CCDSVM has provided concurrent users each from a single web browser anywhere on the network to securely perform permitted various tasks, which could be run concurrent within a single web-browser. Because of console support software **6** of control system **2** let each users have exclusive view for resource objects in their private workspace **16** and in said common workspace **15** of a group each user belong to, and let each user to manage resource objects in their own private space, the multiple users in multiple groups model communication or in 1-to-1 communication can concurrently post or un-post files, folders, messages, or other resource objects without interfere each pair of user in 1-to-1 communication or users between each user-groups. Thus, the workspaces of each user and each group are well protected.

Hence, the console support software **6** of control system **2** provides privileged users to perform tasks of creating multiple users and user-groups on control system, and each group can be assigned with variable size of users for online interactive meeting. The total number of users and the total number of groups, and maximum number of users per groups of WCUWE of CCDSVM can support is an implementation issue and depends on the capacity of the control system **2**. The WCUWE of CCDSVM and CCDSVM has been designed with unlimited scalability for both users and provisioned systems **3**. On the other hand, the spirit of this invention does not limit to with the boundary issues of WCUWE and CCDSVM.

FIG. **8**A and FIG. **8**B illustrate that each user-X and user-Y has a different private workspace than others, and each has assigned with different system resources. Regardless each user is a member of group or without any group membership, the console support software **6** of control system **2** also can

provide user X and user Y to peer-to-peer instantly post files or folders one a time to each other, or instantly un-post the previous posted files and folders through similar steps as posting or un-posting materials in group online meeting model:

1) Let each user can perform clicking on source of file or folder for post or un-post;
2) Let user select post operation task if user want to post any file or folder through operation menu; or let user select un-post operation task from operation menu if said user like to un-post a previously posted file or folder;
3) upon submitting operation task, let each user provide input to prompt of "who is a targeted user?" provided by said console support software **9**, and further submit the selected task;
4) the console support software **6** obtains and parses task information of original user, source of file or folder, type of operation, and other information;
5) If said type of operation is post file or folder, said console support software **6** deposits the information of said source file or folder to said private file and folder area of said private workspace of targeted user. If said operation type is un-post a file or folder, said console support software **6** search a corresponding file or folder information from said private file and folder area of said private workspace of said targeted user to match the source file or folder, if found a match, the corresponding entry of the file or folder is removed from said private file and folder area of said private workspace of targeted user; and
6) Upon each user refreshing the web-page in web-browser **9**, the console support software retrieves and convert the updated information in said private file and folder area of private workspace of targeted user into presentable web-format (web-page) and transmit said web-page to web-browser **9** of each user for displaying and viewing.

For example, as shown in FIG. **8**A, user X can follow said steps 1) to 3) described above to post his "folder **2**" **32** in displayed files and folders area of private working space of user X's web-browser **9** to user Y. As shown in FIG. **8**B, upon user Y refreshing his/her said web-browser **9** on a system by walking on any system node or folder node in displayed resource tree, user Y will see said "folder **2**" **32** in his/her displayed private file and folder tree of private workspace of user Y shown in FIG. **8**C. As matter of the fact, when walking system node and file & folder tree nodes on said web-browser **9**, a refreshing web-page screen request is generated and to be handled as described previously or in steps described above. As a results of participating 1-to-1 mode communication with multiple users, a user, such user X, can view and access multiple folders or files posted by one or more users from each of own private workspace. On the other hand, the user X has no permission to delete any resources posted by other users.

In addition to post messages in group-based communication involving parties with two or more users, WCUWE of CCDSVM also provide each user to post message to each other user with non-group based 1-to-1 message model. FIG. **7**A and FIG. **7**B have shown an example of user X and user Y, who may not be in same user-group, however, the console support software **6** of control system **2** provides each user to share and exchange messages with each other from a three-section web-page in said browser **9**.

The section-3 of web-page in said browser **9** let user to work with same said steps of non-group based 1-to-1 posting files and folders or un-posting previously posted files or folders as described previously. The section-3 included in FIGS. **7**A and **7**B just for a demonstration of how flexible of this invention is to display information under private workspace of each user. Further, the number of sections in the screen layout and the order of each section in web-page display really is not a limitation to this invention.

The section-1 is user message input area together with an additional input field of "To who" due to unlike in group online meeting, the console support software **9** have no knowledge of who is the targeted user in this case. User can posting a message with similar said steps as previous discussed previously of this invention. Nevertheless, unlike with group meeting messaging model, after obtaining and parsing information of the messages from each user, the console support software **6** of control system **2** will deposit the message to both targeted peer user and the original sending user's message areas of private workspace. Further, with same said steps of displaying messages in each user's browser, each user can view both own message and messages from peer user in section-2 of browser.

For example, as shown in FIG. **7**A, user X input a message of "msg**4** of user X" **30** and type "user Y" in input field of "To who", and then submit the message by clicking on "Send" button. Similar as shown in FIG. **7**B, user Y input a message of "Msg**3** of user Y" **31**, and type "user X" in input field of "To who", and then submit the message. Upon the refreshing web-page at web-browser **9** of each user, the user X will see the message **30** he/she sent and the message **31** from user Y. Also the user Y will see both messages **30** and **31** at same time.

Again, the design choice of three section of web-page display and the order of each section is just a preferred example of implementation for non-group based peer user communication. This invention does not limited by web-screen display layout, for example, a screen display only with said section-1 and section-2 are also possible and so on without limitation, and each section can display one or more same or different resources in one or more user private workspaces or group common workspaces.

The WCUWE of CCDSVM has provided a security hierarchy for members of any team to work together cross multiple locations. With said security hierarchy and said steps of posting or un-posting file and folder, the console support software **6** of control system **2** can specially let only team leader with capability of posting folder or file to be viewed, shared, and accessed by all team members or by an individual member in non-group meeting based environment. In addition, console support software also let each team member to post or un-post his/her file or folders to only a specific peer user one a time in non-group-meeting based environment. Again the files or folders could be on said control system **2** or on any provisioned said provisioned system **3**.

It is relatively straightforward for WCUWE of CCDSVM, more specifically the console support software **6**, to support massive online users, who do not have account with CCDSVM and can not login CCDSVM yet can view the dynamically posted files, folders, or messages by users with user account of CCDSVM.

In a preferred example, such as shown in FIG. **6**E, console support software **6** of control system **2** provides a public web-page to massive online users, who do not have to login CCDSVM. The public web-page, in one embodiment, could be layout with two major sections with section-1 to display messages in sharable message area in a designated user-group's common workspace while the section-2 to display file or folders in sharable file or folder area in said designated user-group's common workspace.

For users belong to same said designated user-group, each user from a web-browser on any system of said network to login to CCDSVM and get said group online meeting web-page as shown in FIG. **6**A for user X and FIG. **6**B for user Y. Further as described previously each user in said group can instantly post message to said message area of designated user-group's common workspace, or post one file or folder to the sharable file and folder area in designated user-group's common workspace, which associated with said public web page. Also, each user in user-group can instantly perform un-post the previously posted files or folders task, which results removing the corresponding file or folder entry in file and folder area in designated user-group's common workspace by said console support software **6** of the control system **2**, or un-post the previously posted messages from common message area of common workspace of said user-group.

Meanwhile as previously described and as shown in FIG. **6**E, the public web-page can be displayed in the browser **9**, which may located on one or more public display systems or on each public online user's native system, can be dynamically updated and further to be viewed by the massive online public users with the same pace and synchronized with identical contents as dynamically displaying in said section-2 and section-3 of said web-page/browser **9** displayed in group online meeting as shown in FIG. **6**A and FIG. **6**B. Again, said two sections of web-page display layout is a preferred example and it does not limit this invention because with this invention other layout is also possible, for example, it may be desired to have web-page layout only contains one section, which is either for messages or file or folder's or for both messages or files display depending on the application's need.

As shown in FIG. **8**A, FIG. **8**B, and FIG. **8**C, if people who had participated multiple sessions of group based or non-group based peer-to-peer communication and has posted or un-posted many messages, files and file-folders may often do not remember how many materials has been posted for sharing. For example, as shown in FIG. **8**A, user X can select "folder **2**" **32**, select post on operation menu, and provide target "user Y" to share "folder **2**" **32**. As show in FIG. **8**C, user Y can see "Folder **2**" **32** of user X in web-browser of user Y after user X post "Folder **2**" **32**. If user X select various files or folders for other dozen people to post and share, user X will have difficult time to remember what has been shared and who are targeted users. As shown in FIG. **11**, the console support software **6** of control system **2** can provide each user, user X for example, a central view of a control list of how many messages, files or file-folders being posted by him or her and to who.

As shown in FIG. **11** of personal share management, in view of user X's web-browser **9**, an entry of posted "Folder **2**" **30** is listed at bottom of said control list and it is posted/shared to user Y. Further, said console support software **6** can provide each user selectively un-post the previously posted files and folders by clicking on the "delete shared?" of operation menu for corresponding entries in the shared list of center view web-page and further to generate a un-post operation task. For example, user X clicks on "delete share?" to un-post "Folder **2**" **32**. Thereafter, as described before, said console support software **6** of control system **2** will execute the un-post task. Further upon refreshing the targeted user Y's web-page/web-browser, in stead of seeing "Folder **2**" **32** as shown in FIG. **8**C, user Y will see an updated view of said of web-page as shown in FIG. **8**B, where the "Folder **2**" **30** has been removed. The operation menu for "delete share?" depicted here just for example and it could be displayed in any other form.

In addition to let original user in a group to control the un-posting file or folder in group based communication, the console support software **6** of control system**2** may also provide a privileged user to un-post the previously posted files or file folders by other users in said displayed file and file-folder section of said web-page. To support this capability, said console support software **6** will provide an operation menu to associate with said displayed files and folders in said display section, further to allow privileged user to perform similar un-post operation. After receiving the un-post operation, the console support software **6** will perform similar said searching for selected file or folder in the group file and folder area **19** of FIG. **4**B of group common workspace of said group, if there is match the corresponding entry will be deleted. Also, the file and folder section of web-page/web-browser can be refreshed accordingly as described before.

The present invention can apply to update web contents for massive web portal in enterprises and personal communication. In one example, the FIGS. **6**A and B may represent an internal portal web page, encoded with information in common workspace of the group-I, and the FIG. **6**E web page display may represent a external portal of that group. The present invention has provided synchronization between these two internal and external web pages in real-time.

In one embodiment, a web page screen of a news portal may divided into multi-sections including sports section, medical section, education section, entertainment section, financial section, traveling and vacation section, and technology section etc. without limit. With the CCDSVM, each different section can be assigned to each different corresponding group, which is dedicated for updating the news in the section. For example, 5 sports journalists in a sports group scattering around the world can be assigned to update the sports news section instantly at anytime and anyplace, where the sports section is only associated with sports group's common workspace and can be updated and refreshed independently in the web page via a run-time script. In another example, a financial group has 4 journalists being assigned to update the financial section of the web page, which is only associated with financial group's common workspace, and so forth without limits. Of course, the rest of sections can also be assigned to each dedicated group of journalists.

In another embodiment, a retailer's portal may have multiple sections of web page layout for displaying hundreds or even thousands of products, where each section may be dedicated to each type of products. By assigning each section to a dedicated group of one or more users, the contents of each section can be much secure and efficiently updated because more users can concurrently working on each section of each web page for entire web pages of any portal without interrupt each others.

Further, each section can be dedicated to encode and display one type of resources from each user in corresponding area of group common workspace or encode and display with mixed type of resources of from multiple areas of a group common workspaces such as mixed with photo-picture and text messages and others, or mixed with streaming video and text messages.

Instead of encoding resources of one or more users' private workspaces into each corresponding section of a web page indirectly via each corresponding group's common workspace, in which each user are the members as described in prior embodiment, each section of a web page can be encoded directly with resources information in one or more users' private workspaces. To accomplish this, the console support software modules **6** of the control system **2** can keep a mapping list between each section and the associated one or more

users' private workspaces for each web page. In addition, the console support software **6** can facilitate each user an alternative post operation and an alternative un-post operation to support updating the contents of the assigned section of a web page. Therefore, instead of facilitating each user posting resources information into the user's group's workspace or withdrawing resource information from the user's group's workspace, it facilitating user to tag the resource in his private workspace for posting directly to the assigned section, or to tag the previously tagged of posted resource for withdrawing. In one embodiment, the tagged information can be kept in a list and be store in a dedicated place in the user's private workspace, or a central controlled file-folder that console support software **6** can easy to pick up. Therefore, during generating the targeted web page, it will encoding each section of web page according to the mapped information combined with each user's tagged information for updating web contents. Although, this is not as robust as post and un-post operations, yet it reduced the usage for group common workspace.

The management for larger volumes of web contents for a complex web portal will become much efficient because it divides the larger amount of works into one or more small jobs for each corresponding small section of each web page of a web portal. Therefore, each section can be parallel updated by one or more users yet the updating is independent and without interfering other sections. Second, it is much secure since each section is restricted to be updated by limited users. Also, with this invention, the updating does not require any hardcode for each section due to the nature of posting and withdrawing technology introduced in present invention.

Having briefly described embodiments of the web based network environment in which the present invention operates, FIG. **12** illustrates an example of a computer system **1200**, which is an exemplary client system **10**, provisioned system **3**, control system **2**, or console system **1** in which the features of the present invention may be implemented. Process of the web based information exchange can be implemented in any processor-based computer system, such as a PC, a workstation, or a mainframe computer. It will be apparent to those of ordinary skill in the art that other alternative computer system architectures may also be employed.

Referring back to FIG. **12**, computer system **200** includes a processing unit **201**, an interface bus **211**, and an input/output ("IO") unit **220**. Processing unit **201** includes a processor **202**, a main memory **204**, a system bus **211**, a static memory device **206**, a bus control unit **205**, a mass storage memory **207**, and a web information exchange module **230**. Bus **211** is used to transmit information between various components and processor **202** for data processing. Processor **202** may be any of a wide variety of general-purpose processors or microprocessors such as Pentium™ microprocessor, Motorola™ 68040, or Power PC™ microprocessor.

Main memory **204**, which may include multiple levels of cache memories, stores frequently used data and instructions. Main memory **204** may be RAM (random access memory), MRAM (magnetic RAM), or flash memory. Static memory **206** may be a ROM (read-only memory), which is coupled to bus **211**, for storing static information and/or instructions. Bus control unit **205** is coupled to buses **211**-**212** and controls which component, such as main memory **204** or processor **202**, can use the bus. Bus control unit **205** manages the communications between bus **211** and bus **212**. Mass storage memory **207**, which may be a magnetic disk, an optical disk, hard disk drive, floppy disk, CD-ROM, and/or flash memories for storing large amounts of data. Web based information exchange **230**, in one embodiment, is a network control component for facilitating information exchange over the Web. It should be noted that web based information exchange **230** could be software or hardware or a combination of software and hardware components and modules. In one embodiment, it comprises video and audio recoding and playback components including digital camera or video or audio encoding and playing back PCI cards such that the on-the-fly recoded streaming video on the computer system **200** can be displayed into each user's web browser or other viewing mechanism on another remote computer system **200** via communication link over various network during online information exchanging.

I/O unit **220**, in one embodiment, includes a display **221**, keyboard **222**, cursor control device **223**, and communication device **225**. Display device **221** may be a liquid crystal device, cathode ray tube ("CRT"), touch-screen display, or other suitable display device. Display **221** projects or displays images of a graphical planning board. Keyboard **222** may be a conventional alphanumeric input device for communicating information between computer system **200** and computer operator(s). Another type of user input device is cursor control device **223**, such as a conventional mouse, touch mouse, trackball, or other type of cursor for communicating information between system **200** and user(s).

Communication device **225** is coupled to bus **211** for accessing information from remote computers or servers, such as server **104** or other computers, through wide-area network **102**. Communication device **225** may include a modem or a network interface device, or other similar devices that facilitate communication between computer **200** and the network. Computer system **200** may be coupled to a number of servers via a network infrastructure such as the infrastructure illustrated in FIG. **1**.

The present invention has been described in considerable details with reference to certain examples. However, other versions and examples are also possible, therefore, the spirit of this invention shall not be limited to these examples and/or embodiments. Also, the claims of this invention will label with numbering and in most cases, the numbering does not represent any sequence or order unless specifically described as the steps of a method.

What is claimed is:

1. A computing device comprising a processor, memory and program code which, when executed by the processor, configures the device to:

(i) display a user interface to each of a first user and a second user to share information, wherein each of the user interfaces comprises, for each of the first and second users, (a) a private section configured to display information about files or folders available for the user to share and (b) a common section configured to display information about files or folders shared with the user;

(ii) share a file or folder selected, from the available files or folders, by the first user with the second user by (a) allowing the first user to identify the file or folder in the private section on the first user's interface, which is not viewable by the second user, (b) unlocking a protection mechanism of the file or folder to allow access to the second user, (c) storing information about the file or folder, without the content of the file or folder, in a common work place accessible to both the first user and the second user, and (d) displaying information about the file or folder in the common section on the second user's interface, wherein the second user can access the file or folder through the displayed information; and

(iii) stop sharing of a file or folder to the second user that the first user has previously shared with the second user by (a) deleting information about the file or folder dis-

played in the common section on the second user's interface, (b) deleting information about the file or folder that has been stored in the common work place, and (c) locking the protection mechanism to rescind access to the second user.

2. The computing device of claim **1**, wherein each of the user interfaces further comprises, for each of the first and second users, (c) a private message section configured to allow the user to input a message and (d) a common message section configured to display messages shared with the user.

3. The computing device of claim **2**, wherein the program code further configures the device to

(iv) share a message by the first user with the second user by (a) receiving a message from the first user, via the private message section on the first user's interface, which is not viewable by the second user, (b) storing the message in the common work place accessible to both the first user and the second user, and (c) displaying the message in the common message section on the second user's interface; and

(v) delete a message that the first user has previously shared with the second user by (a) identifying the message in the common work place and in the common message section on the second user's interface and (b) deleting the message from the common work place and the common message section.

4. The computing device of claim **1**, wherein the user interface is presented on a web browser.

5. The computing device of claim **1**, wherein the first user and the second user are members of a user group, and the file or folder shared with the second user is also displayed in the common section of the user interfaces of all users in the user group, and is accessible to all users of the user group.

* * * * *

Exhibit C

US008713442B2

(12) **United States Patent**
Tsao

(10) **Patent No.: US 8,713,442 B2**
(45) **Date of Patent: Apr. 29, 2014**

(54) **METHOD AND APPARATUS FOR INFORMATION EXCHANGE OVER A WEB BASED ENVIRONMENT**

(75) Inventor: **Sheng Tai (Ted) Tsao**, San Jose, CA (US)

(73) Assignee: **Sheng Tai Tsao**, Fremont, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 192 days.

(21) Appl. No.: **13/080,007**

(22) Filed: **Apr. 5, 2011**

(65) **Prior Publication Data**

US 2012/0084658 A1 Apr. 5, 2012

Related U.S. Application Data

(60) Division of application No. 11/732,496, filed on Apr. 2, 2007, and a continuation-in-part of application No. 10/713,904, filed on Aug. 6, 2002, now Pat. No. 7,418,702, and a continuation-in-part of application No. 10/713,905, filed on Aug. 12, 2002, now Pat. No. 7,379,990.

(60) Provisional application No. 60/787,653, filed on Mar. 31, 2006.

(51) **Int. Cl.**
G06F 3/14 (2006.01)

(52) **U.S. Cl.**
USPC **715/736**; 715/760; 709/201; 709/203; 709/205; 707/740

(58) **Field of Classification Search**
USPC 715/200–277, 700–867; 700/701–866; 709/201–229; 705/50–79; 345/30–111
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,865,545	B1 *	1/2011	Estrada et al.	709/201
2011/0185292	A1 *	7/2011	Chawla et al.	715/760

* cited by examiner

Primary Examiner — Ruay Ho
(74) *Attorney, Agent, or Firm* — James Zhu, Jun He Law Office, PC

(57) **ABSTRACT**

Sharing and exchanging information via world wide web is a way of life in modern society. The dynamic workspace technology together with customized posting and removing operations provide people a much efficiently and flexibly communication platform for sharing and exchanging information across a global environment.

20 Claims, 18 Drawing Sheets

An example of resource objectives on systems and can be assigned to each group or each user' work space.

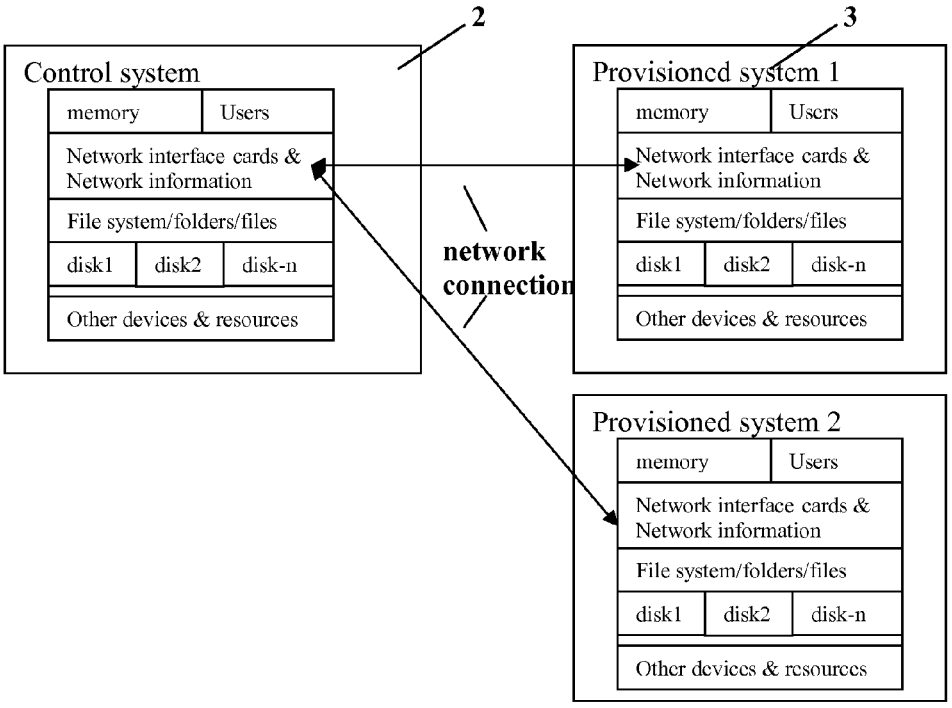


Fig. 1: A Typical CCDSVM platform with Multiple Clients

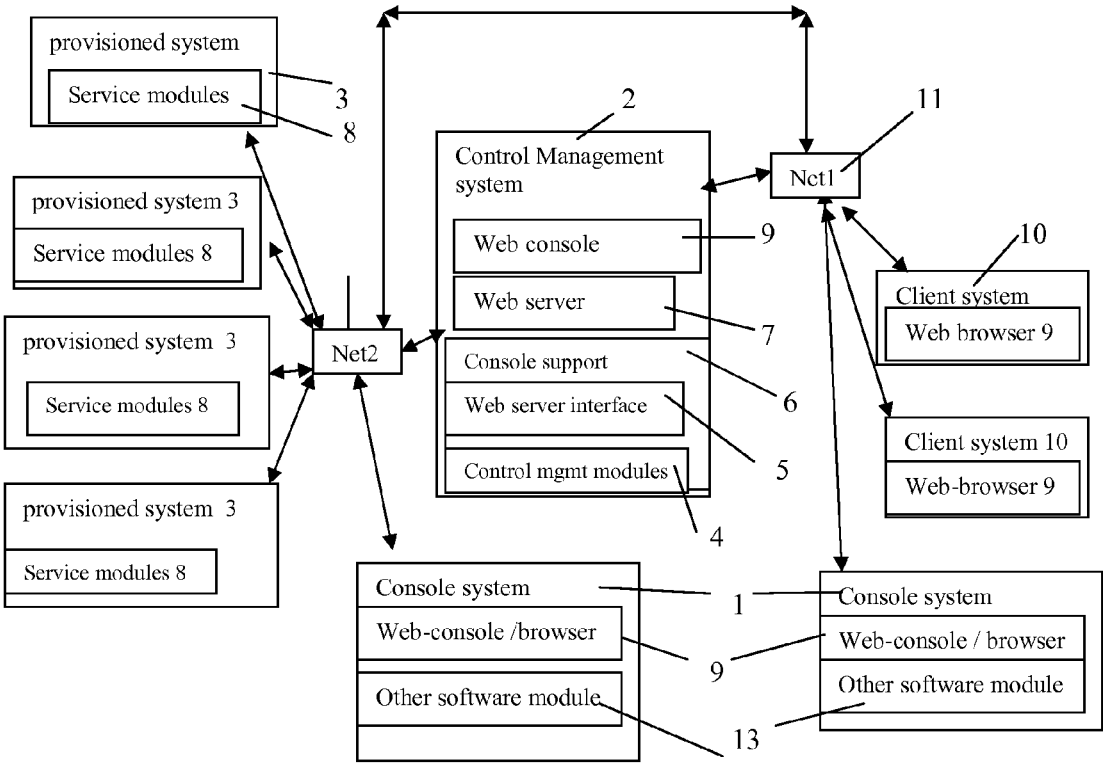


Fig. 1

Fig. 2A: A variation of the CCDSVM platform for web-browser based communication:

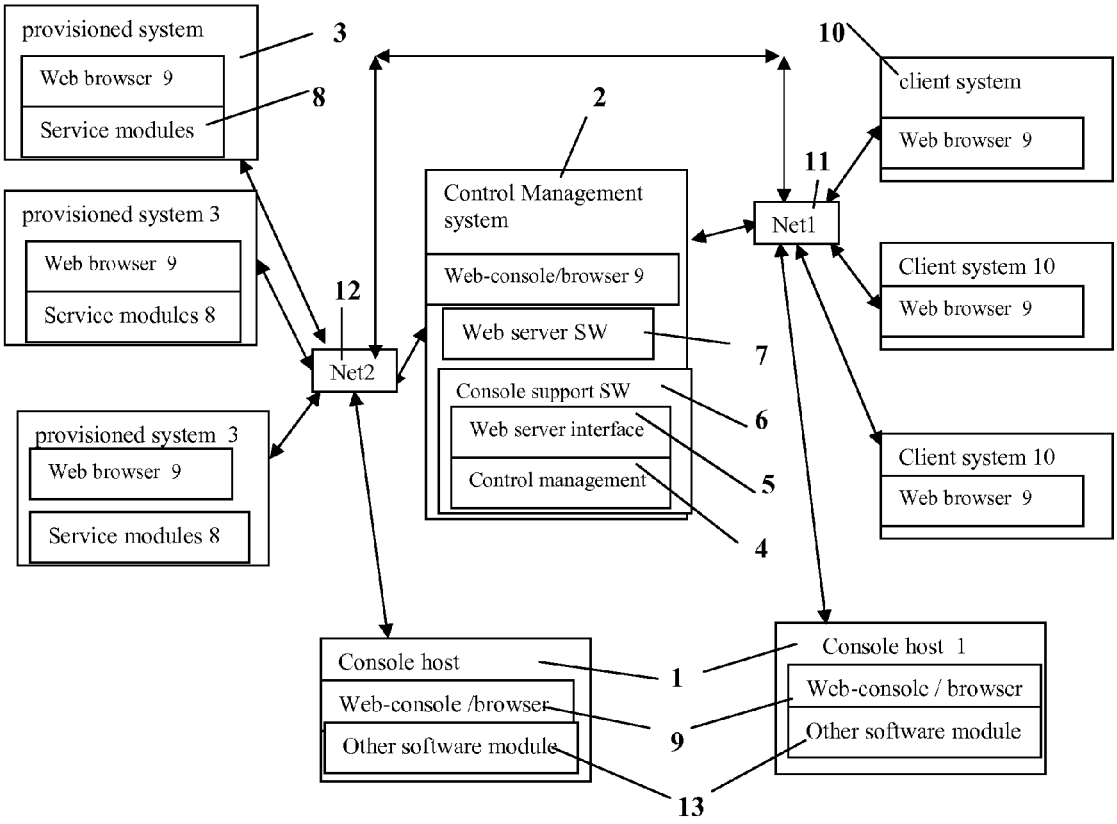


Fig. 2A

Fig. 2B: A variation of CCSDVM platform, the degenerated CCDSVM with only control system and without provisioned systems, and to be accessed by massive client systems.

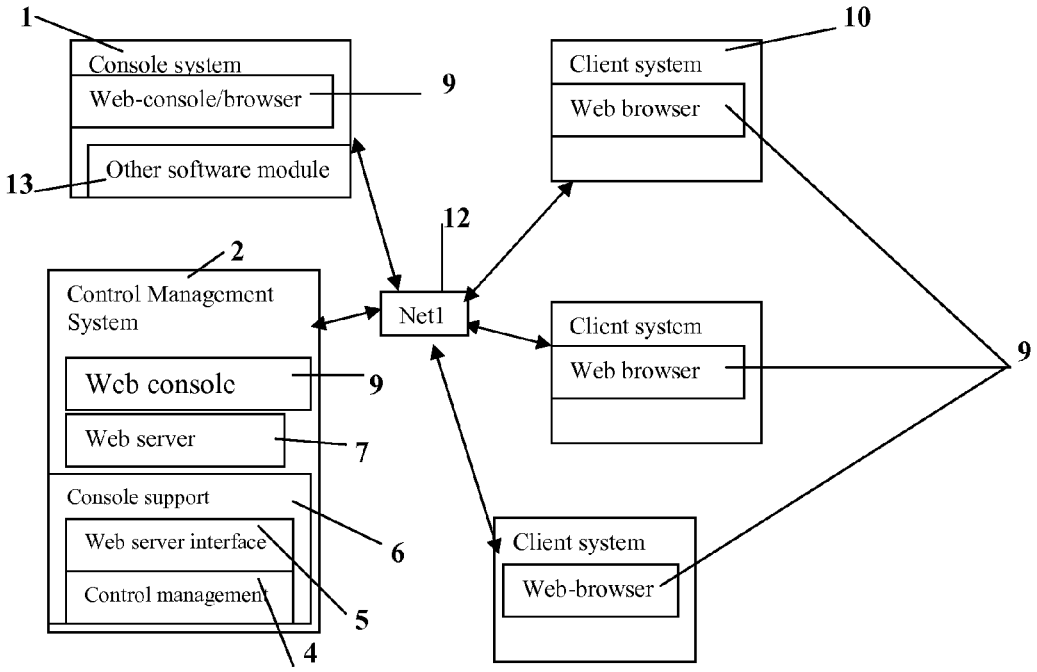


Fig. 2B

Fig. 3: The Software of WCUWE for typical CCDSVM:

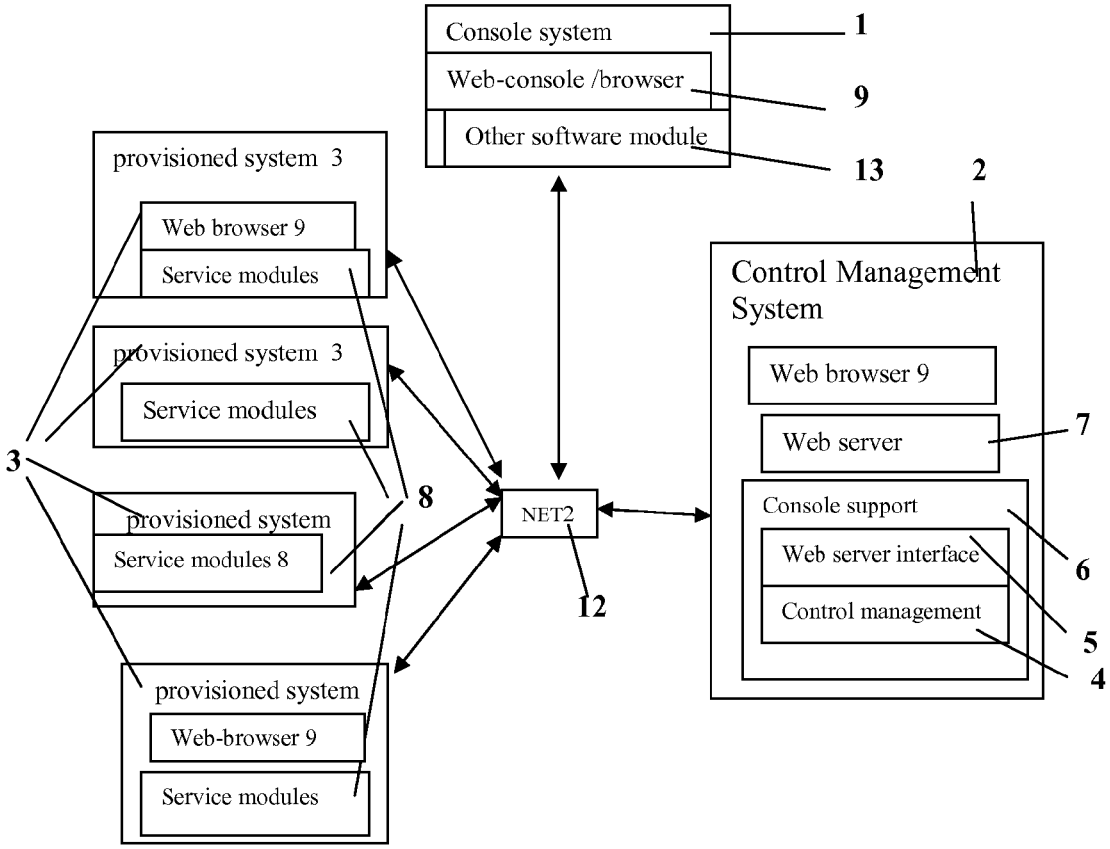


Fig. 3

Fig. 4A: Dynamic work space (central location for share) on Control system.

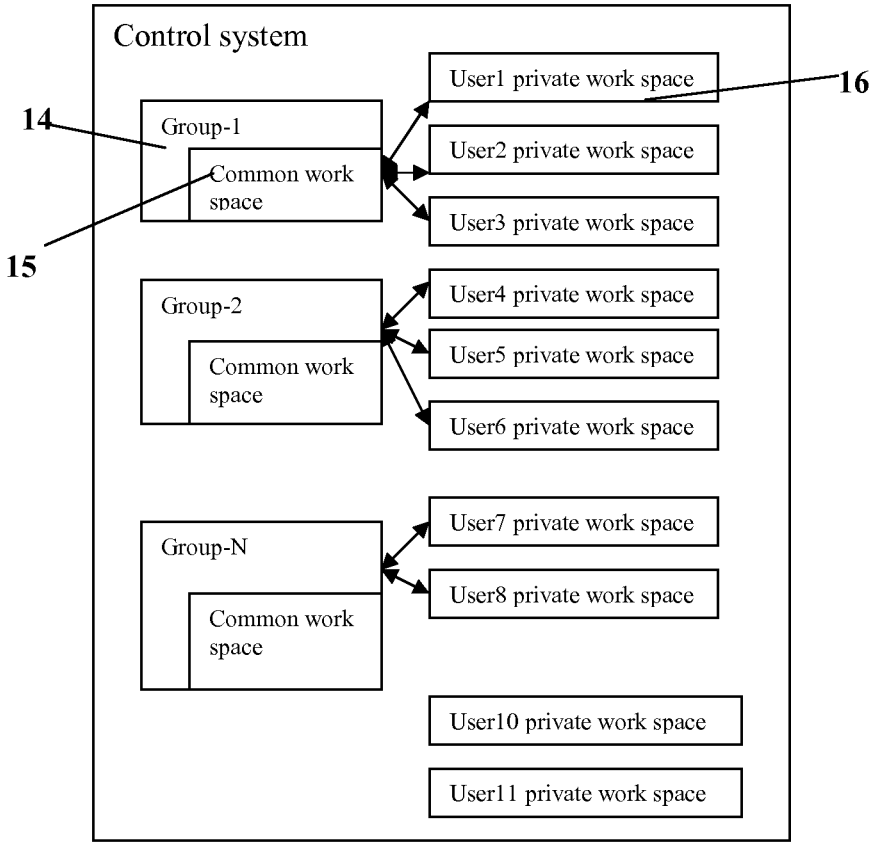


Fig. 4A

Fig. 4B: an example of resource may be assigned to a user-group common work space.

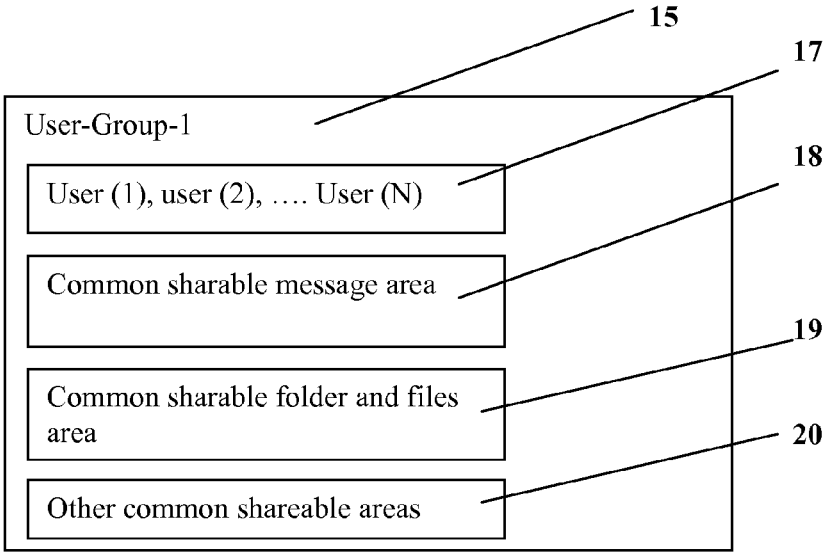


Fig. 4B

Fig. 4C: an example of user private work space.

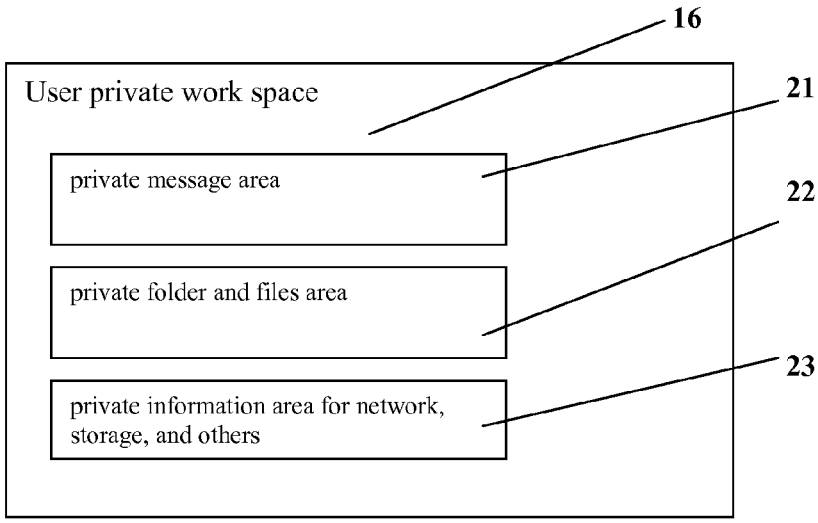


Fig. 4C

Fig. 5: An example of resource objectives on systems and can be assigned to each group or each user' work space.

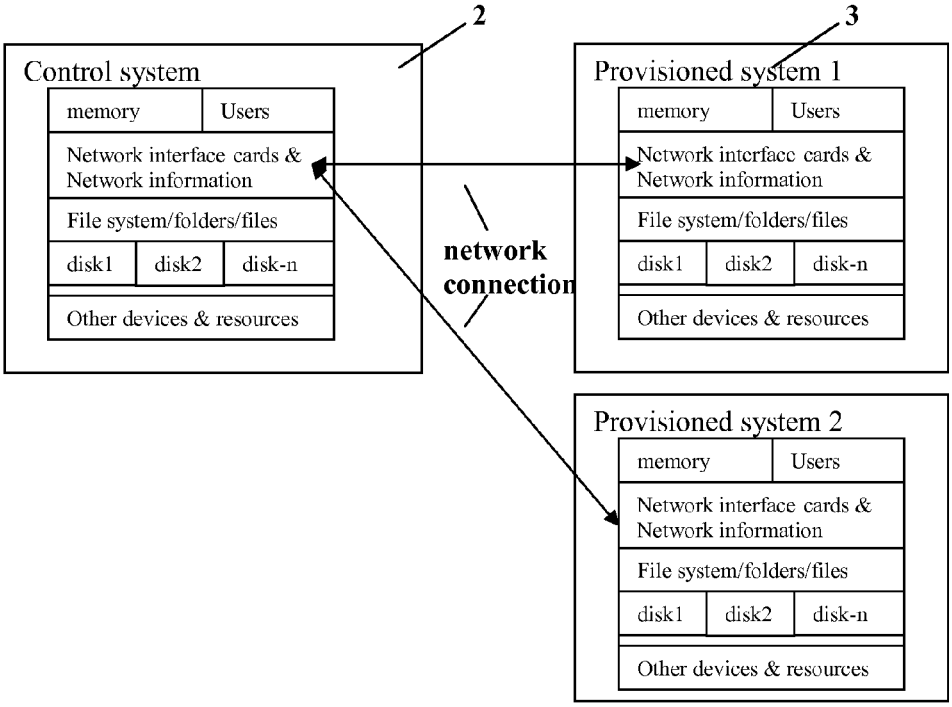


Fig. 5

Fig. 6A: An example of displaying a web-page with 4 sections in web-browser of user X in user-group-1 **during** an interactive online meeting.

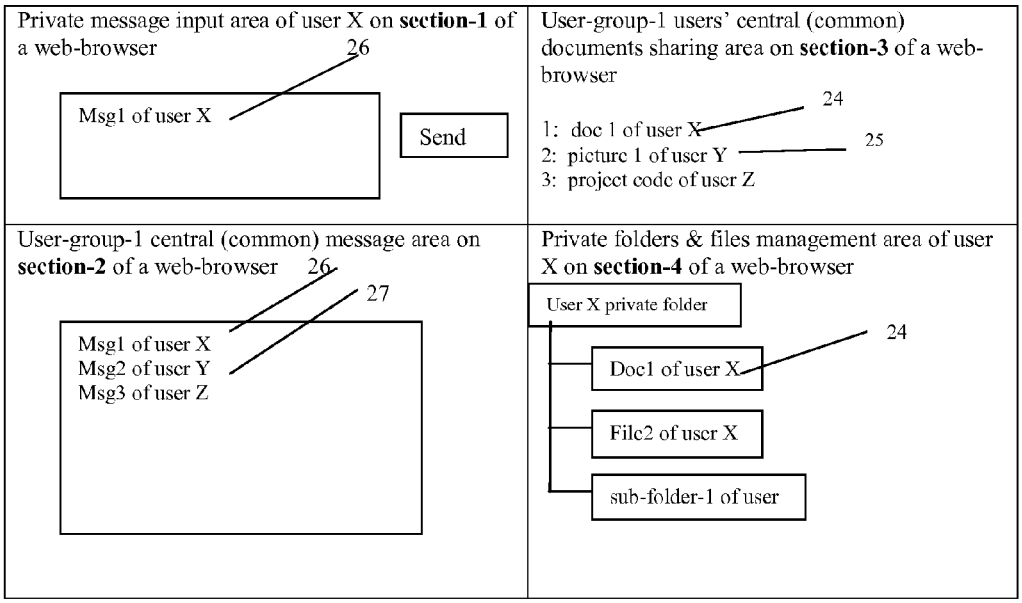


Fig. 6A

Fig. 6B: An example of displaying a web-page with 4 sections in web-browser of user Y in user-group-1 **during** interactive online meeting.

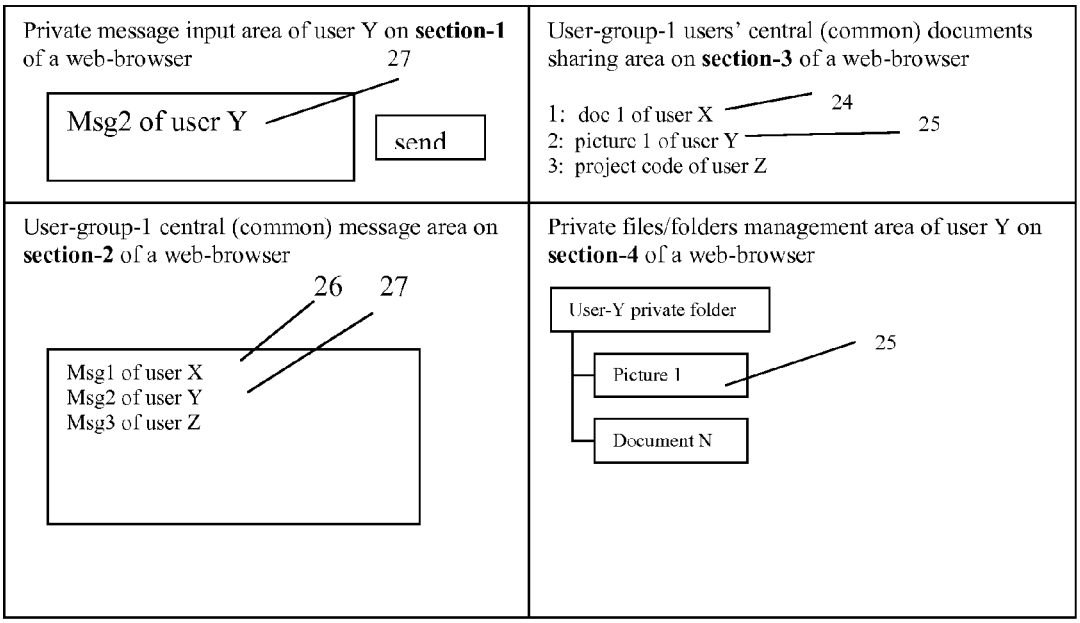


Fig. 6B

Fig. 6C. An example of displaying a web page in a web browser for user X in user group-1 **before** interactive online meeting.

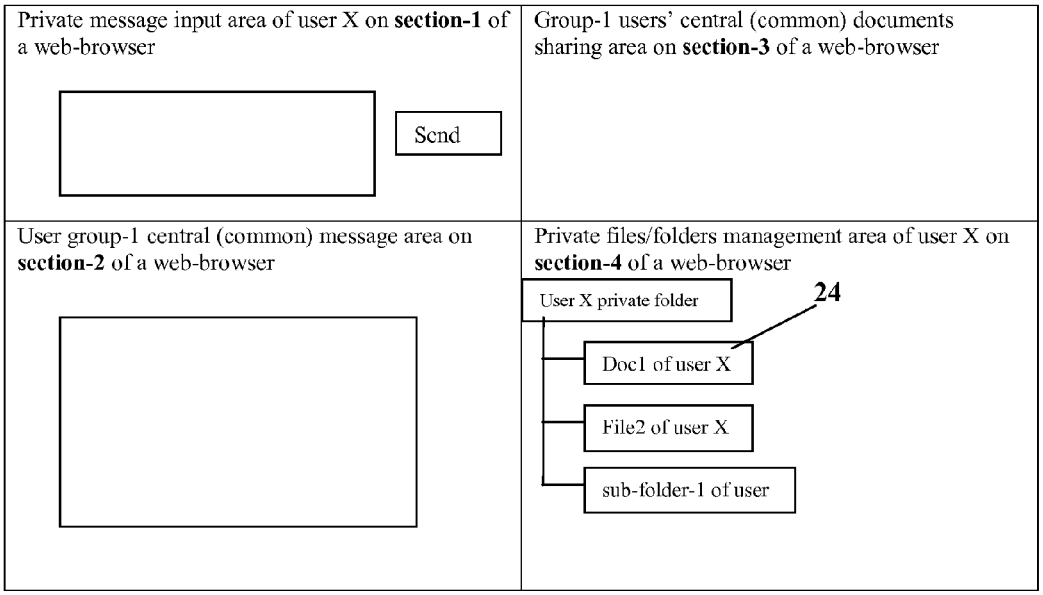


Fig. 6D: An example of display a web-page with 4 sections in browser of user Y in user group-1 **before** interactive online meeting.

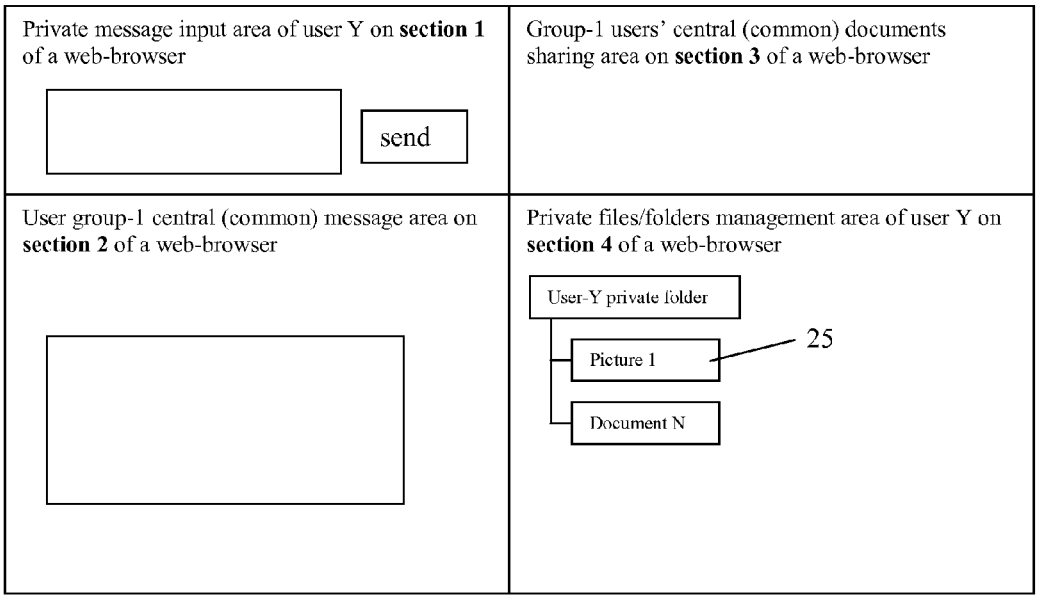


Fig. 6E: An example of displayed web-page that each massive online user can access and view from each of their browser without login to CCDSVM during said user X and user Y in user-group-1 online meeting.

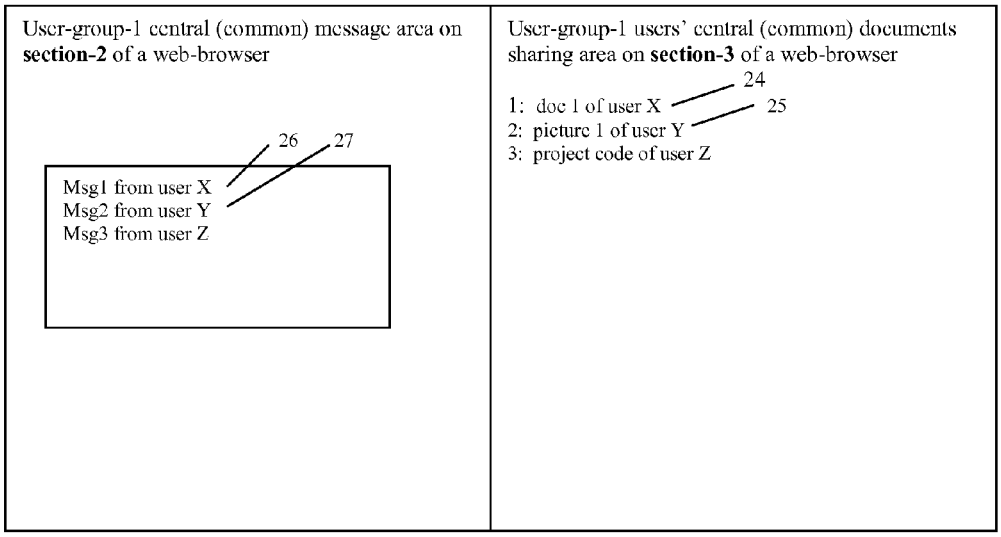


Fig. 6E

Fig. 7A: Non-group based one-to-one peer-to-peer post messages and folders of user X

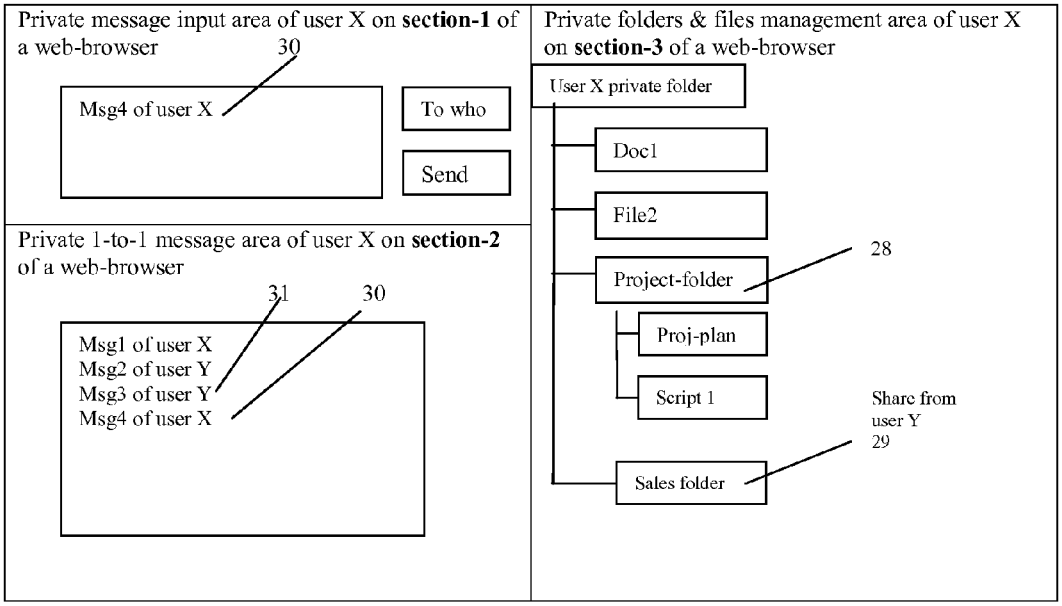


Fig. 7A

Fig. 7B: Non-group based one-to-one peer-to-peer post messages and folders of user Y

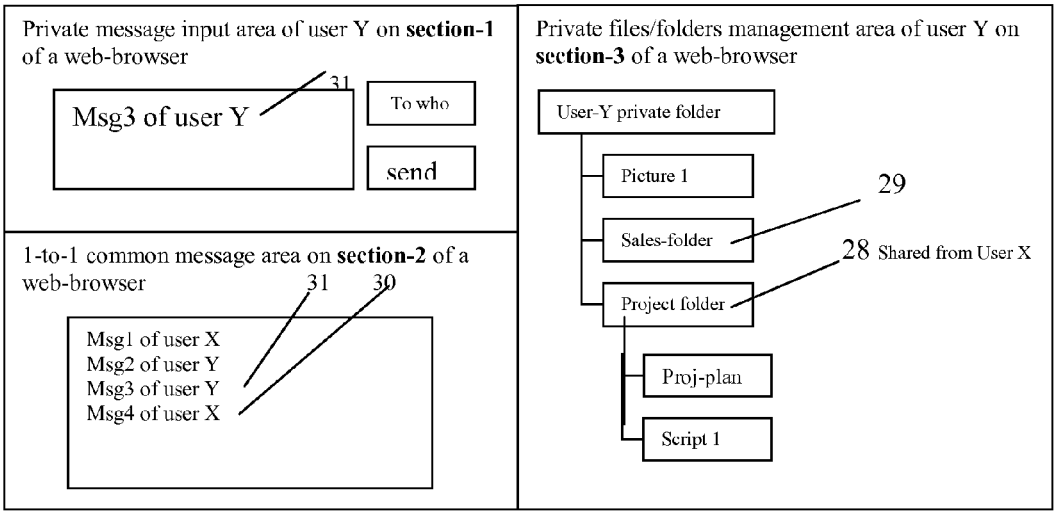


Fig. 7B

Fig. 8A: An example web graphic presentation view of assigned resources in a user X's private work space.

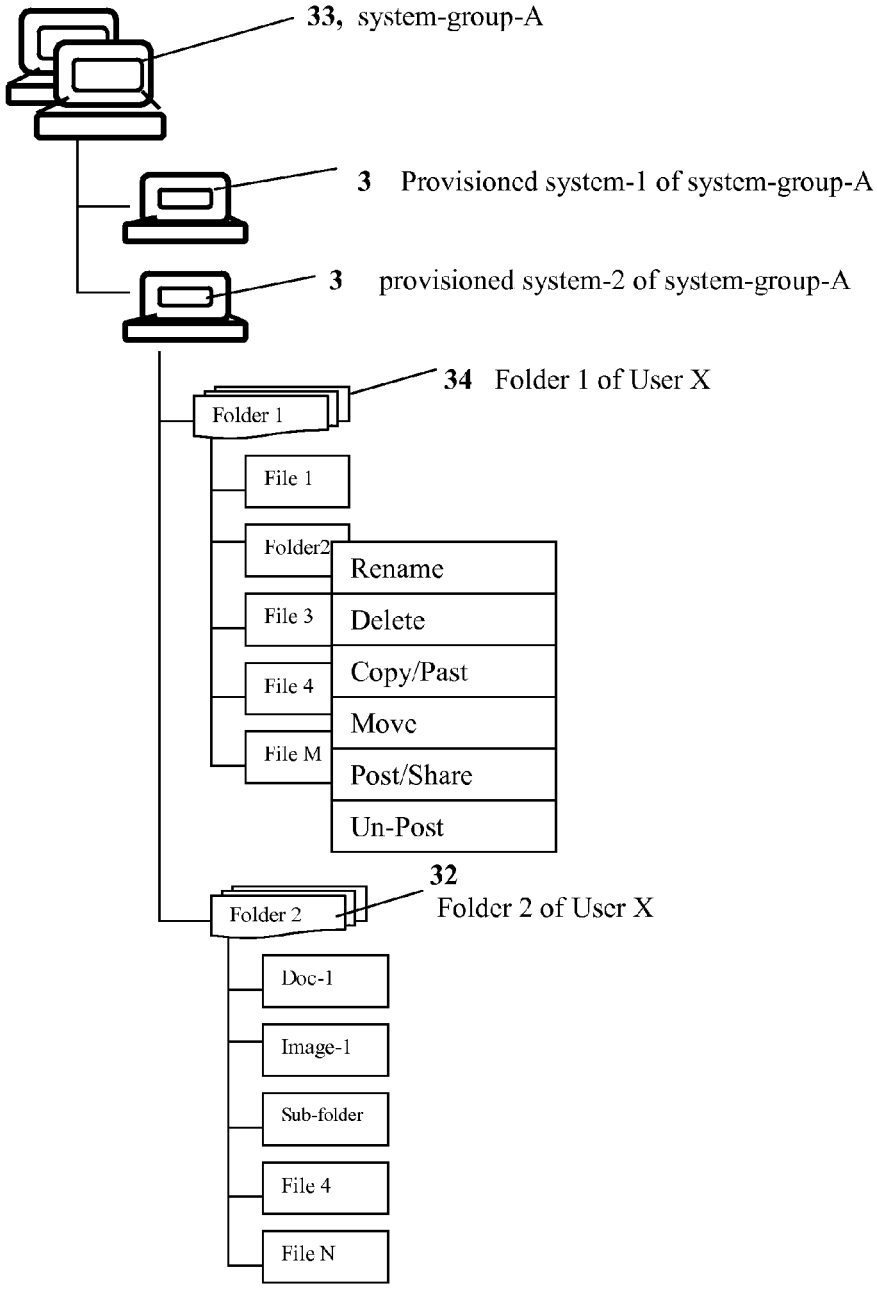


Fig. 8A

Fig. 8B examples of web-based graphic presentation of assigned resource in the private work space of user Y.

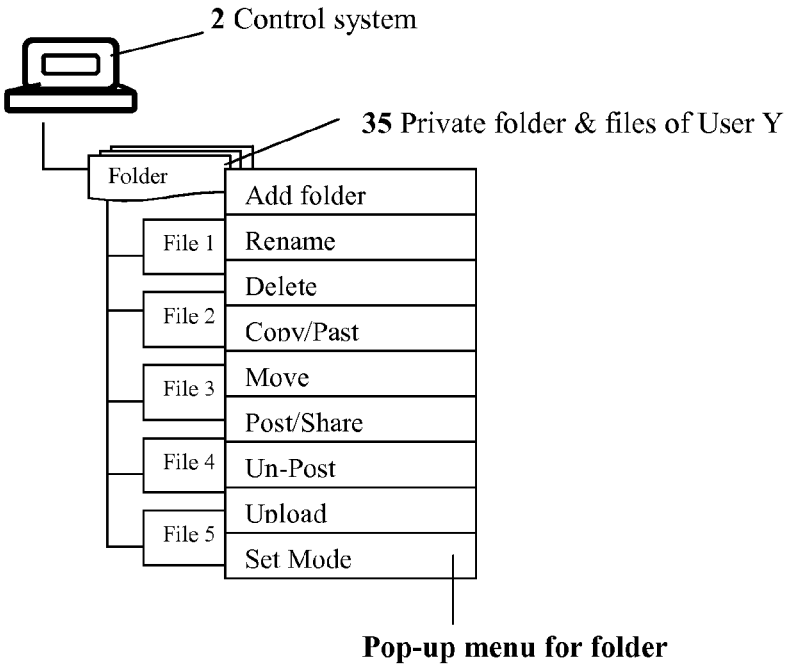


Fig. 8B

Fig. 8C: An example web graphic presentation view of user Y's private work space after user X dynamically posted and shared "folder 2" in user X's private space.

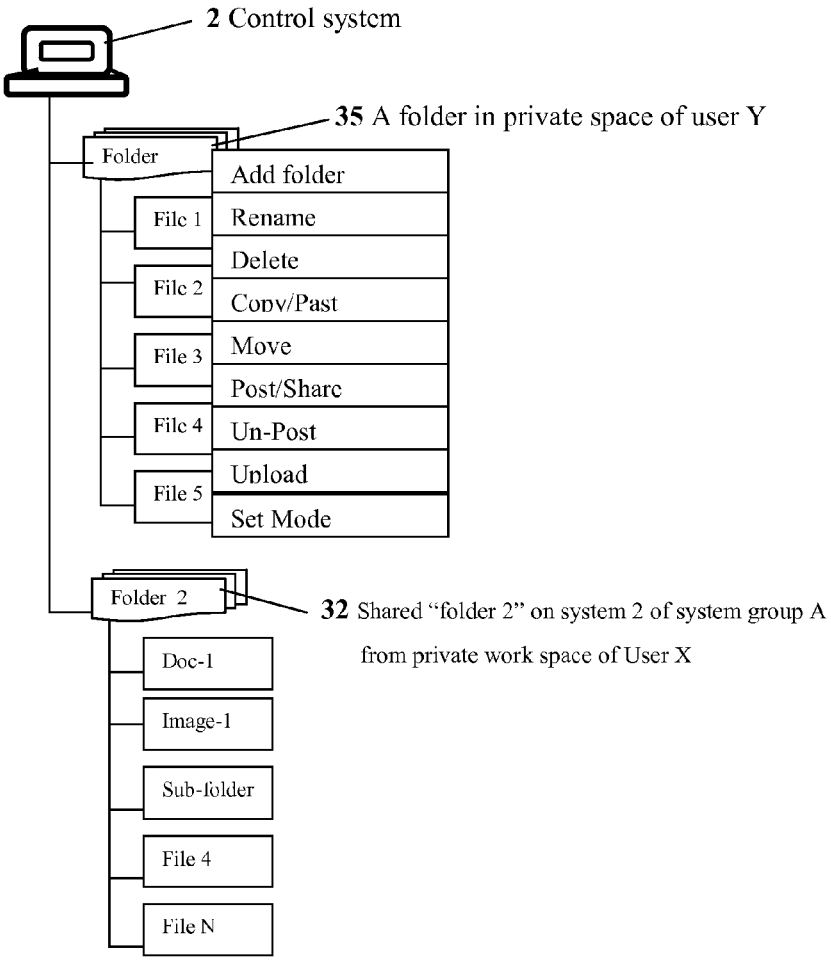


Fig. 8C

Fig. 9A: an example of pop-up menu for folder

Add folder
Rename
Delete
Copy/Past
Move
Post/Share
Un-Post
Unload
Set Mode

Fig. 9B an example of pop-up menu for file

Rename
Delete
Copy/Past
Move
Post/Share
Un-Post
Unload
Set Mode

Fig. 10A an example of non pop-up operation menu for folder:

Add folder	rename	delete	Copy/past	move	Post/share	Un-post	upload	Set mode

Fig. 10B an example of non pop-up operation menu for folder:

rename	delete	Copy/past	move	Post/share	Un-post	Set mode

Fig. 11: an example of center view of list of posted files or folders in file & folder area of a private work space of a user X in a special personal share management section of a web-page.

File or Folder Name	On System	Share to Who	
My-video jpg file	System-1	Jane	Delete shared?
Project plan	Engineer-3	Jim	Delete shared?
C program code for web	Support-2	John	Delete shared?
Picture of building	System-1	Jane	Delete shared?
Engineer Drawing	Engineer-3	Jim	Delete shared?
Patent application-v1	System-1	Ted	Delete shared?
Folder 2	System 2 of system group-A	User Y	Delete shared?

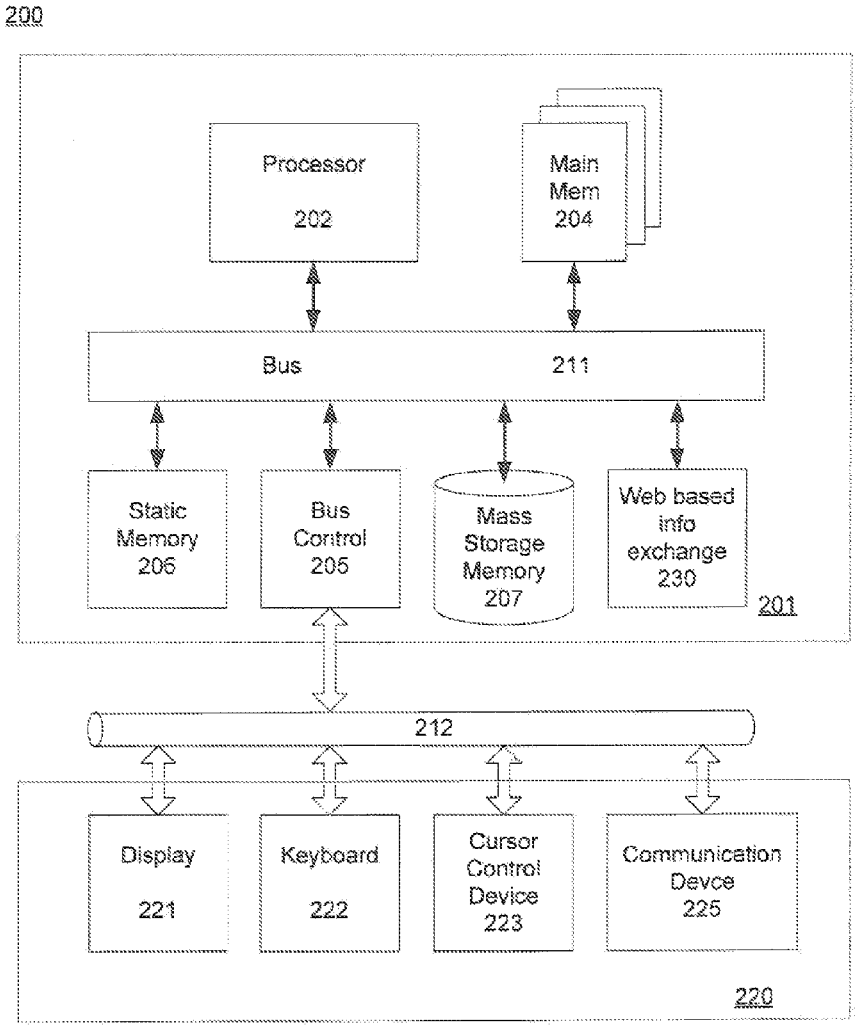

200
Processor
202
Main
Mem
204
Bus
211
Static
Memory
206
Bus
Control
205
Mass
Storage
Memory
207
Web based
info
exchange
230
201
212
Display
221
Keyboard
222
Cursor
Control
Device
223
Communication
Devce
225
220

FIG 12

METHOD AND APPARATUS FOR INFORMATION EXCHANGE OVER A WEB BASED ENVIRONMENT

PRIORITY

This application is a divisional application for U.S. patent application Ser. No. 11/732,496 of filed on Apr. 2, 2007. The application Ser. No. 11/732,496 has claimed the benefit of the priority of the U.S. provisional application No. 60/787,653. The application Ser. No. 11/732,496 is also the continuation-in-part of U.S. patent application Ser. No. 10/713,904, filed on Aug. 6, 2002 and converted from the U.S. provisional application No. 60/401,238. The application Ser. No. 11/732,496 is also a continuation-in part of U.S. patent application Ser. No. 10/713,905, filed on Aug. 12, 2002 and converted from the U.S. provisional application Ser. No. 60/402,626. The patent application Ser. No. 10/713,904 now is a U.S. Pat. No. 7,418,702. The patent application Ser. No. 10/713,905 now is a U.S. Pat. No. 7,373,990. All of the above prior applications are incorporated herein by reference in each one's entirety.

FIELD OF THE INVENTION

The present invention generally relates to communication network. More specifically, the present invention relates to web based communication system.

BACKGROUND OF THE INVENTION

With increasing popularity of using the Internet and World Wide Web ("the Web") for the rapidly changing digital world, individuals as well as enterprises, exchange (or swap) large volumes of information through the Web. The demand for larger and faster information exchange has increasingly grown in recent years. For example, various business meetings and conferences are conducted over the Web with attendees scattered around the world. To enhance the communication between the meeting attendees, not only voice (and/or image) information is important, but information exchange can also be critical.

Fast, volume, and secure information exchange is important in enhancing the efficiency of a communication network. A problem associated with a conventional network is limited size and speed. For example, a typical email system limits the size of each email, such as 10 megabytes, thereby a reasonable speed of delivery of such email can be achieved. Another 1 method and apparatus for information exchange over a web based environment May 3, 2013 problem associated with a conventional network is that various hosts require software installation before a user can use their platform. For example, Skype™ requires a user to install its proprietary software on the user's systems before it allows the user to use its platform.

Accordingly, there is a need in the art to provide a faster, more secure, volume information exchange over the Web.

SUMMARY OF THE INVENTION

The present invention relates to a method and apparatus for information exchange over a web environment.

With the development of central controlled distributed scalable virtual machine (CCDSVM) and the web-based computer user working (operating) environment (WCUWE), the problems mentioned in the previous section can easily be solved by introducing a "dynamic work space" technology of this invention within the frameworks of the CCDSVM and the WCUWE.

With the dynamic work space technology, one or more login users of the CCDSVM, each using a browser residing in any computing system from anywhere, can instantly perform various tasks. For example, each user can post or un-post messages, files, folders, or other resources residing in a plurality of computing systems of the CCDSVM in group model or in 1-to-1 personal model to one or more other user across a network. This technology will also achieve larger scales of secure information exchange without size limitation and installation of special software.

The implementation of this invention has provided users with a web-browser based, 1-click system that is capable of securing exchanges of messages, files, folders, or other resources residing in computing systems crossing network domains without size limitation.

Additional features and benefits of the present invention will become apparent from the detailed description, figures, and claims set forth below.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be understood more fully from the detailed description given below and from the accompanying drawings of various embodiments of the invention, which, however, should not be taken to limit the invention to the specific embodiments, but are for explanation and understanding only.

FIG. **1** is a block diagram illustrating a layout of a CCDSVM with a central control management system, a plurality of provisioned systems, client systems, and console systems in accordance with one embodiment of the present invention;

FIG. **2**A is a block diagram illustrating a layout of a CCDSVM with which users from their web-browser of the provisioned system can communicate with each other in accordance with another embodiment of the present invention;

FIG. **2**B is a block diagram illustrating a layout of a CCDSVM with only a single control management system to be accessed by a plurality of client systems in accordance with another embodiment of the present invention;

FIG. **3** is a block diagram illustrating an example of a software structure of the WCUWE for a CCDSVM in accordance with one embodiment of the present invention;

FIG. **4**A illustrates an example of a dynamic work space in accordance with one embodiment of the present invention;

FIG. **4**B illustrates an example of possible resources which may be assigned to a user-group common work space in accordance with one embodiment of the present invention;

FIG. **4**C illustrates an example of a user private work space and possible assigned system resources in accordance with one embodiment of the present invention;

FIG. **5** illustrates an example of resources residing in computing systems in accordance with one embodiment of the present invention;

FIG. **6**A illustrates an example of displaying a web-page with 4 sections via a web-browser of a user X in a user-group during an interactive online meeting in accordance with one embodiment of the present invention;

FIG. **6**B illustrates an example of displaying a web-page with 4 sections via a web-browser of a user Y in a user-group-**1** during an interactive online meeting in accordance with another embodiment of the present invention;

FIG. **6**C illustrates an example of displaying a web-page with 4 sections via a web-browser of a user X in a user group-**1** before the interactive online meeting in accordance with another embodiment of the present invention;

FIG. **6**D illustrates an example of displaying a web-page with 4 sections via a web-browser of a user Y in a user group-**1** before the interactive online meeting in accordance with another embodiment of the present invention;

FIG. **6**E illustrates an example of a displayed web-page that each of a plurality of massive online users can access and view from each of their browser screens without login to the CCDSVM during said user X and user Y in the user-group-**1** online meeting in accordance with another embodiment of the present invention;

FIG. **7**A illustrates one-to-one (peer-to-peer) post or un-post messages, files and folders by an user X in accordance with one embodiment of the present invention;

FIG. **7**B illustrates one-to-one (peer-to-peer) post or un-post messages, files and folders by an user Y in accordance with another embodiment of the present invention;

FIG. **8**A is an example of a web-based graphic presentation of assigned resources in a user X's private work space in accordance with one embodiment of the present invention;

FIG. **8**B is an example of a web-based graphic presentation of assigned resources in the private work space of an user Y in accordance with another embodiment of the present invention;

FIG. **8**C is an example of a web-based graphic presentation of user Y's private work space after the user X dynamically posted and shared a "folder **2**" in the user X's private space in accordance with another embodiment of the present invention;

FIG. **9**A is an example of a pop-up menu for operating folders in accordance with one embodiment of the present invention;

FIG. **9**B is an example of a pop-up menu for operating files in accordance with another embodiment of the present invention;

FIG. **10**A is an example of a non pop-up operation menu for operating folder in accordance with one embodiment of the present invention;

FIG. **10**B is an example of a non pop-up operation menu for operating file in accordance with another embodiment of the present invention;

FIG. **11** is an example of a displayed list of posted files or folders in a sharable file & folder area of a common work space of a user X in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION

Embodiments of the present invention are described herein in the context of a method, system and apparatus for providing a web based communications network with fast, volume, and secure information exchange. Those of ordinary skill in the art will realize that the following detailed description of the present invention is illustrative only and is not intended to be in any way of limiting. Other embodiments of the present invention will readily suggest themselves to such skilled persons having the benefit of this disclosure. Reference will now be made in detail to implementations of the present invention as illustrated in the accompanying drawings. The same reference indicators will be used throughout the drawings and the following detailed description to refer to the same or like parts.

In the interest of clarity, not all of the routine features of the implementations described herein are shown and described. It will, of course, be appreciated that in the development of any such actual implementation, numerous implementation-specific decisions must be made in order to achieve the developer's specific goals, such as compliance with application- and business-related constraints, and that these specific goals will vary from one implementation to another and from one developer to another. Moreover, it will be appreciated that such a development effort might be complex and time-consuming, but would nevertheless be a routine undertaking of engineering for those of ordinary skill in the art having the benefit of this disclosure.

In accordance with the present invention, the components, process steps, and/or data structures described herein may be implemented using various types of operating systems, computing platforms, computer programs, and/or general purpose machines. In addition, those of ordinary skill in the art will recognize that devices of a less general purpose nature, such as hardwired devices, field programmable gate arrays (FPGAs), application specific integrated circuits (ASICs), or the like, may also be used without departing from the scope and spirit of the inventive concepts disclosed herein. Where a method comprising a series of process steps is implemented by a computer or a machine and those process steps can be stored as a series of instructions readable by the machine, they may be stored on a tangible medium such as a computer memory device (e.g., ROM (Read Only Memory), PROM (Programmable Read Only Memory), EEPROM (Electrically Eraseable Programmable Read Only Memory), FLASH Memory, Jump Drive, and the like), magnetic storage medium (e.g., tape, magnetic disk drive, and the like), optical storage medium (e.g., CD-ROM, DVD-ROM, paper card and paper tape, and the like) and other known types of program memory.

Those of ordinary skill in the art will realize that the following detailed description of the present invention is illustrative only and is not intended to be in any way limiting. Other embodiments of the present invention will readily suggest themselves to such skilled persons having the benefit of this disclosure. It will be apparent to one skilled in the art that these specific details may not be required to practice to present invention. In other instances, well-known circuits and devices are shown in block diagram form to avoid obscuring the present invention. In the following description of the embodiments, substantially the same parts are denoted by the same reference numerals.

While particular embodiments of the present invention have been shown and described, it will now be apparent to those skilled in the art having the benefit of this disclosure that many more modifications than mentioned above are possible without departing from the inventive concepts disclosed herein. Therefore, the appended claims are intended to encompass within their scope all such modifications as are within the spirit and scope of this invention.

The present invention can be implemented by a web based system operated in a network infrastructure, such as a central controlled distributed scalable virtual machine ("CCDSVM"). FIG. **1** illustrates a CCDSVM in accordance with an embodiment of the present invention. The CCDSVM includes a group of provisioned systems **3**, a control management system **2**, console systems **1**, and client systems **10**. The control management system ("control system") **2** is configured to control a group of systems ("provisioned system") **3**. The console systems ("console systems") **1** provide management for the CCDSVM while the client systems **10** ("client systems") initiate access to the provisioned systems **3** and/or

the control system **2** via a web-browser across a network infrastructure, such as an Intranet, the Internet, and/or a local area network ("LAN").

The network infrastructure ("network") includes all necessary hardware and software that facilitate various users from different geographic locations to communicate through the Internet, an Intranet, and/or a LAN. The hardware of the network includes wired network connection media, wireless connecting media, circuitry components, and communication equipments. The network connection media includes various cables such as Ethernet and/or optical fiber. The wireless connecting media is capable of establishing wireless communication links through air. The circuitry components include processor, data bus, memory devices, and/or circuit board. The communication equipments, such as switches, routers, gateways, and/or adapters, are used to provide a communications network. It should be noted that there are other possible elements of communication equipment that may be needed to form a communication link, but they are not necessary to understand the present invention.

The software infrastructure includes Internet Protocol ("IP") addresses and system name identification software such as Domain Name Server ("DNS"), firewall software, IP gateway set-up software, IP broadcast, and so forth. The communication protocols over the network could be IP-based standard or non-standard proprietary protocols such as Hypertext Transfer Protocol ("HTTP") over Transmission Control Protocol/Internet Protocol ("TCP/ IP") or Simple Object Access Protocol ("SOAP") over TCP/IP. It should be further noted that some IP based protocols are proprietary protocols and some are non-IP based protocols.

A web-browser is an application program that allows a user access to information across the Web. The web browser could be, for example, a Windows Internet Explorer ("IE"), FireFox, Netscape, and/or Mozillar. A web-browser could also be a proprietary software, which uses web-protocols such as HTTP, WAP, SOAP for communication across the network. The web-browser can be implemented with any suitable programming languages such as C, C++, Java, or XML, or a combination of the suitable languages.

A web page, on the other hand, is encoded with information displayable through the web-browser. The web page may be hosted by a web server in a web server system and can be sent or retrieved by using the web-browser.

The provisioned systems **3**, client systems **10**, console systems **1**, and control management system **2** are computational systems that could be a server system, desktop or laptop system, handheld devices such as PDA, wireless phone, and so forth. The computational systems include CPUs, storage devices, and memory controller. The storage devices include internal memory such as read-only memory ("ROM") and/or random access memory ("RAM"). The storage devices may further include external storage devices such as magnetic disks or tapes, which include redundant array of inexpensive drives ("RAID"), just bunch of disk drive ("JBOD") and memory stick. The associated storage controller can be Integrated Drive Electronics ("IDE"), Small Computer System Interface ("SCSI"), Fiber optical controller, or a combination of above-mentioned devices.

The computational systems also include a native non-embedded or real-time embedded OS, which could be Linux, or Windows, Unix, or proprietary OS. It should be noted that the present invention may include multiple control systems at different layers in a system configuration topology wherein each of the control systems is capable of communicating with each other through a network. The control systems at middle layers play dual roles as both a control system and a provisioned system in the system configuration topology.

FIG. **1** illustrates a CCDSVM platform with multiple clients in accordance with one embodiment of the present invention. The CCDSVM includes the console systems **1**, control management system **2**, provisioned systems **3**, and networks **11-12**, wherein the networks **11-12** are used inter-connecting with all systems of the CCDSVM and the client systems **10**.

The console system **1** is a computational system having a web-browser ("web-console") **9** wherein the web-browser **9** is used by privileged users for accessing the CCDSVM platform. The web browser **9**, which could reside in any computing system, permits a user access to information in the CCDSVM by following, in one example, a web URL link. For example, a privileged user enters a web URL of "https://69.107.28.123/stt/sttwebos" on a command line of web browser **9** screen and once the link is established, the user can obtain information hosted by the web-based computer user work/operation environment ("WCUWE"). The console system **1**, in one embodiment, further includes software modules **13**, wherein the software modules **13** may be used to facilitate communication between the console system **1** and the control management system **2**.

The control system **2**, in one embodiment, is a server system, a desktop or a laptop computer with the memory and storage capabilities. The control system **2** includes a web server software **7** and a console supporting software **6**. The console supporting software **6** includes web server interface software modules **5** and control management software modules **4**. The control management system **2** may also have web-browser **9** used as a web-console **9** of the control system **2**. The web server software **7** sends/receives data to/from the web-console **9** of the console system **1** or the client system **10** or the control system **2**. The web server software **7** is configured to provide secure sockets layer ("SSL") encryption for encrypting the data before its transmission, thus to enhance the security. The web server software **7** could be commercially available software such as Apache™ from open source, or IIS from Microsoft or a proprietary software. The web server software modules **7** and console support **6** software modules **6** can be implemented by any suitable or a combination of any suitable programming languages such as C, C++, Java, JavaScript, HTML, or XML.

The console supporting software **6** communicates with the service software modules **8** of each of the provisioned systems **3**. The web server interface **5** of the console support software **6** is capable of providing special functions that are otherwise performed by the web server **7**. If no provisioned system **3** is coupled to the CCDSVM platform, the control management system **2**, in one embodiment, is reconfigured to continue monitoring systems over the network and the CCDSVM platform is also reconfigured to become a single stand-alone system, as shown in FIG. **2**B. Users can access and/or obtain data objects or system resources via the web-browser **9**, which can be resided at any client system **10** or console system **1**.

The provisioned system **3** is a computational system, which comprises suitable native operating system ("OS"). The provisioned systems **3** can be automatically provisioned by the control system **2** through an automatic system service pool construction protocol. The Provision process, also known as pooling process, is a method allowing the control system **2** to control one or more network systems **3** by reconfiguring the network systems. For example, when a provisioned system **3** boots up via a communication protocol, the control system **2** obtains provisioned system's name, IP address, and other system information, wherein the system

information includes network information, storage information, file system information and so forth. Each provisioned system **3** can be monitored, accessed, and/or operated by a user(s) through the web-browser **9** of a console system **1**. In an alternative embodiment, provisional system **3** is operated by users at the client systems **10** via a control system **2** with proper user' authentication. Each provisioned system **3** contains service software modules **8**, which is used to communicate with servers across across a network. For example, the service software modules **8** of the provisioned system **3** can communicate with the control management software modules **4** of the control system **2** to carry out tasks for viewing or operating resources of the provisioned system **3**. The service software **8** can also communicate with the web-browser **9** of the client system **10** or console system **1** to transfer data between them or deliver service to them, or to communicate with another provisioned system **3** to send or receive data.

The service software modules **8** may include special software modules having compatible functionalities of the web-server software **7** of the control system **2** and to dedicate handling HTTP protocol or other web protocols if there is needs for web-based communication with the client **10** or with other provisioned systems **3** or with the control system **2**. The software modules **8** could be commercially available web server software **7** or any proprietary software. The service software modules **8** could be implemented with any suitable programming languages such as C, C++, Java, or JavaScript.

Net **11** and Net **12** are the network infrastructures that are capable of providing communication links between the control management system **2**, console systems **1**, client systems **10**, and/or provisioned systems **3** without limits.

The client systems **10** may not be a part of the CCDSVM, but with the permission and authorization, users from the web-browser **9** of the client system **10** can login to the CCDSVM and access the permitted resources of the CCDSVM by following a web URL link of the CCDSVM. While the privileged users can obtain, manage, access, and/or operate system resources of the CCDSVM through the WCUWE, regular (non-privileged) users at the client systems **10** may be permitted to access only limited system resources of the CCDSVM.

The CCDSVM configuration, in one embodiment, includes four data flow paths. The first data path is a communication links between the web browser **9** of the client host **10** (or console host **1** or control management system **2**) and the web handling software on the control system **2** (such as the web-server **7** and/or console support software **6**). With this path of data flow, whenever a user sends a request from the web-browser **9** to the web-server **7** and further passed to the console support software **6**, the console support software **6** collects required information from each target systems and converts them into standard structured information (web-format) for web communication. The targeted system could be one of the provisioned systems **3** or the control system **2**. The information collected by the console support software **6** of the control system **2** could be the system status, storage information, network information, user authentication profile, file system information, or files & folders information on the control system **2**, or on any targeted provisioned system **3**. The console support software **6** then passes this converted structured information to web server software **7** and further transmits it to the web-browser **9** through communication link Net **11** and/or Net **12** so that the collected information can be displayed and viewed by a conventional web browser **9**, as shown in FIG. **1**.

The communication protocols used between the web-browser **9** of the client system **10** (or the console system **1** or the control system **2**) and the web server **7** of the control system **2** could be HTTP, HTTPS (SSL encrypted HTTP protocol) or any suitable web protocols for web communication, which could successfully send or receive the data across the world wide web, or could also be other standard or proprietary IP-based on non-IP-based protocols. This data path may be referred to as the console support software **6** transmits data to or receives data from the web-browser **9**.

The second data path is the data flow through communication links between the control system **2** and the provisioned systems **3**. Requests targeted to a provisioned system **3** are passed from the console support software **6** of the control system **2** to the service modules **8** of the provisioned systems **3** through communication link network **12**. When those requests are received by the targeted provisioned system, the service modules **8** of the provisioned system **3** carry out the requested tasks, and then send a response back to the console support software **6** of the control system **2**. The implementation of the actual products of this invention with the second data path may use proprietary TCP/UDP/IP based protocols for the communication between the provisioned systems **3** and the control system **2**. However, other standards or proprietary IP-based or any suitable non-IP-based protocol are also possible. The communication protocols used between the console support software **6** of the control system **2** and the service modules **8** of the provisioned system **3** can be any suitable IP based or non-IP based protocols, whichever is suitable to transmit data between them. The typical data flow through this path could be the boot message, system status, network information, or storage information of the provisioned system **3** without limits.

The third data path is the data flow between the provisioned systems **3** and the client system **10** or the console system **1** via the communication link **11**-**12**. Referring back to FIG. **1**, the console support software **6** of the control system **2** may present a web link, which points to an object on a provisioned system **3**, to a user working on a web-browser **9** of the client system **10** or the console system **1** or the control system **2**. The object pointed by the web link resides in the provisioned system **3** could be a text file, MPEG video, PDF document, MS Power Point, Word documentation, etc. It also could be a link of another web service program. From the web browser **9**, a user can directly access the information residing in a provisioned system **3** pointed by the web link without go through the control system **2** again. In this case, the service modules **8** of the provisioned system **3** also include the web server software modules **7** or equivalent one to directly support the web browser **9** as mentioned before. The communication protocols with the third path could be the web protocols such as HTTP or WAP. Alternatively, the communication protocols can be other standards or proprietary IP-based or non-IP-based protocols. To simplify the discussion, the data transmitted on this path will be mentioned as the service software module **8** that send data to or receive data from the web-browser **9** and vice versa.

The fourth data path is data flow through communication link between the provisioned system **3** and another provisioned system **3**. With this path, the service modules **8** of one provisioned system **3** can directly transmit data or information to the service modules **8** of another provisioned system **3** via a communication link provided by the Net **12** without going through the control system **2**. The implementation of the actual products of this invention, for example, uses a proprietary IP-based protocol for communications between the two provisioned systems **3**. The principle of this invention, however, does not rely on or limit to proprietary protocol. It should be noted that other standard or proprietary

IP-based or any suitable non-IP-based protocol may also be possible. The data and information transmitted through the fourth data path can be various types. For example, a user through a web-browser **9** may open a file folder on a provisioned system **3** and the user can further instruct to transfer a file from the current provisioned system **3** to another provisioned system **3** by a mouse click. The data file will then be transferred directly between the two provisioned systems **3** without going through the control system **2**.

FIG. **2**A illustrates a variation of the CCDSVM platform for the web browser-based communication in accordance with one embodiment of the present invention. The CCDSVM platform illustrated in FIG. **2**A is similar to the platform illustrated in FIG. **1** except every provisioned systems **3** includes a web-browser **9**. Thus, with proper authentication process, each user of the provisioned systems **3** may access, manage their own provisioned system **3** from a web-browser **9** located either on their own local provisioned system **3** or on any client system **10** anywhere across a network. It should be noted that when a user is registered with a provisioned system **3**, the user is authorized to access provisioned system **3** locally but not access the CCDSVM. On the other hand, when a user is registered with the CCDSVM, the user may be then authorized to access one or multiple provisioned systems **3** and the control system **2** of the CCSDVM.

FIG. **2**B illustrates another example of a variation of the CCDSVM, in which there is no provisioned systems **3** in the CCDSVM. Thus the Control system **2** can run on its own and the CCDSVM has degenerated into a single stand-alone system (FIG. **2**B). Thus, users each through a web-browser **9** on any client system **3** (or on console system **1**, or on stand-alone control system **2**) is allowed to access and operate the permitted data objects and system resources on the stand-alone control system **2**.

Unless specifically specified, the console support software modules **6** of the control system **2**, the provisioned systems **3**, the service software modules **8** of provisioned system **3**, and the web-browser **9** on the control system **2**, the client system **10**, and the console system **1** all together are referred to the CCDSVM platform illustrated in FIG. **1**.

A web-based computer user work/operation environment ("WCUWE") of the CCDSVM is a software infrastructure, as illustrated in FIG. **3**. The WCUWE provides each permitted user to log into the CCDSVM and further to provide each login user a private web-browser based user operating environment, which is a private work space including the assigned and permitted resources on the control system **2** or on the provisioned system **3** across a network. The WCUWE further provides each user from a single web-browser to perform various permitted tasks and operations over various system resources (as shown in FIG. **5**), further, the tasks can be performed concurrently from a single login web-browser.

The mentioned tasks include allowing privileged users to create users, groups, assign users to each group, to assign initial associated system resources to each user or group, and to save the information of created users and groups into a database stored on a storage media of the control system. The information needed for creating users or groups include user names, user IDs, assigned security permissions, passwords, access points, group names, group IDs, maximum members in a group, and so forth. The tasks also include various system operations such as manage network, storage, file system, folder, files, messages, user security, and so forth.

The database is organized as one or more lists of tables, wherein each of the lists contains various resources. For example, the resources could be one or multiple users information, one or multiple systems information, one or more network cards, and one or more disk drives information. The storage media could be a system's internal storage or disk drive, for example.

The WCUWE of the CCDSVM includes the software modules of present invention. For example, the software includes the console support modules **6** of the control system **2**, the service software modules **8** of the provisioned system **3**, the other service software **13** of the console system **1**. The current implementation of the WCUWE is based on a proprietary design of this invention. However, the nature and spirit of this invention does not limit to proprietary designed software. The WCUWE of the CCDSVM also includes other proprietary software modules or conventional software such as the web-server modules **7** of the control system **2**, and the web-browser **9** of the control system **2**, the client system **10**, and console system **1**.

The WCUWE of the CCDSVM can be implemented with any suitable or combination of various suitable programming languages such C, C++, Java, JavaScripts, HTML, XML, and so on. To simplify the discussion, the WCUWE will be used to represent software modules in the CCDSVM described above.

Users of the CCDSVM may be created by the CCDSVM though a given process. Also, any users on the Internet, Intranet, or LAN without registering with the CCDSVM may also access the CCDSVM. The WCUWE of the CCDSVM provides user to access various resources and perform various tasks, for example, through a preferred model of mouse clicking on graphically represented resource objects displayed in a web browser **9** screen.

The WCUWE is designed to be operated on a CCDSVM platform, and specially is designed to provide dynamic work space (FIG. **4**A) to each user associated with various CCDSVM. The WCUWE further supports tasks of instantly posting or un-posting user's private resources information to peer members in a group or to each peer user of the CCDSVM not in a group or not in same group, or to massive public users on the network, where the massive public users may do not have accounts with the CCDSVM platform.

The dynamic work space of the WCUWE is a centrally controlled collection of each group's work space and/or user's private work space. The console support software **6** of the control system **2** of the CCDSVM will assign each user a private work space and each group a common work space when each user or group account is created by a privileged user after the corresponding WCUWE software modules configured in the control system **2** and in the provisioned system **3** enter into an operational mode. When a user or a group account is created, each user or group is assigned with specific security scopes and permissions. An important aspect of the WCUWE of the CCDSVM is that it assigns each user and/or each user-group a working space based on user and group security permission.

A work space is an organized information structure, which can be saved in storage media of memory or disk drives or both of the memory and disk drives by the console support software modules **6**. A work space is created during a privileged user performing the tasks of creating user or group. The work space will be initiated and assigned with system resources of the CCDSVM as shown in FIG. **5**. Various system resources either in each provisioned system **3** and/or in a control system **2** include respective memory, system-groups, computing systems in a system-group, file systems, file folders, files, storage media, network media and so on without limits. The common work space **15**, as shown in FIG. **4**B for each group may be initially assigned with information of system resources (illustrated in FIG. **5**) such as users in this

group, messages, file systems, folders and files on a specific provisioned system **3** or on the control system **2**, and so on without limits. The resources of the common work space may also be dynamically assigned or allocated by permitted user after creating group account. For example, during an online interactive meeting, a user posts a file to the common work space, so that a new entry of a file node is added to a file and folder tree area of the common work space. In addition, each group may be assigned with one or multiple users, wherein each user can also be assigned with a private work space **16** as shown in FIG. **4**A and FIG. **4**C during the user account creation. The private work space may also be initially assigned with system resources of the CCDSVM during the user account creation. Alternatively, the private work space is dynamically assigned with the resources of the CCDSVM after the user account being created.

For example, a privileged user can dynamically permit or deny a general user to access a list of storage devices. During the operation, at one second, a user may see the list of from the user's web-browser, and at another second, the user can no longer see the list of the storage devices. Also, if an authorized user is permitted access to a list of storage devices on a provisioned system **3**, and the user further, can take one of the devices offline so that the list of the storage device configured in the provisioned system will have one less devices to be displayed.

In another example, at one second, a privileged user allows a general user to access a file J in a folder N on a system M of the CCDSVM, and at another second, the privileged user resets the previously granted permission, thereby, the general user can view and access the file J in the folder N on the system M of the CCDSVM at one time, and a second later, the general user can no longer view and access the file J. The actual implementation of the console support software **6** of control system **2** has achieved this capability.

In addition, each type of resources in the work space is organized as a list of tables and can be stored in storage media in the control system **2**. The integrity of security of each group is protected by limiting different resources to different groups. As such, each group is authorized to have a limited view over assigned system resources residing in the provisioned system **3** as well as in the control system **2**. The same security integrity at user level can be achieved by assigning different system resources of the provisioned system **3** and/or the control system **2** to different individual user based on assigned security permissions. Therefore, each user is authorized to access and view the assigned resources.

FIG. **4**B has shown an example of assigned resources including users **17**, message area **18**, folders and files area **19**, and others area **20** for other resources in the common work space **15** of a user group. The messages from each user in the user-group's common message area **18**, and the information of the folders and files in the common files and folder area **19** can be shared by all users in the same user group. All resources in each group's common work space can be selectively displayed through the web-browser **9** on each login user's local system, who belongs to a same user group after the resources in said group work space being converted to presentable web-format (web-page) by the console support software **6** of the control system **2** of the CCDSVM.

FIG. **4**C has shown an example of assigned resources in the private work space (**16**) for a user, which includes resources in private message area **21**, private file and older area **22**, and private network or storage information or other information area **23**. The assigned resources only can be exclusively viewed and accessed by the designated user unless he/she agrees to share with peer user crossing group boundary or with peer user in same group. The folder (directory) is a data structure which can hold one more files and sub-folders at logical file system level and at physical level on storage media.

FIG. **8**A has shown an example of displaying an actually implemented portion of a web-page encoded with assigned resources in the private work space **16** of an user X, which includes a system-group-A **33** on said network, multiple provisioned systems **3** under the system-group-A, private folders **32** and **34** of the user X resided on a system, and files under the folders **32** in a displayed private work space **16** of the user X.

FIG. **8**B has shown another example of displaying an actually implemented portion of a web-page encoded with assigned resources in the private work space **16** of an user Y, which includes a control system **2**, private folder and files **35** that belong to the user Y in the private work space **16** of the user Y.

The private work spaces of user X and user Y may also include private message area **21** and other resources, which are not displayed in FIG. **8**A & FIG. **8**B because the WCUWE is designed to dynamically display part of each user's private work space **16** whenever there is needs.

It is notable that said user X and user Y, may be in same user-group such as said group-**1**. However, each of them could have a significant different scope of an assigned private work space **16** assigned by the console support software **6** of the control system **2** due to they have been assigned with very different resources on the systems depending on each user's roles and security permissions, which are setup by said privileged user of the CCDSVM through web-based user interfaces that are provided by the console support software **6**.

The FIG. **6**C and FIG. **6**D are examples of web-page displaying in the web-browser **9** of the user X and user Y in a user-group-**1** before an interactive online meeting. As shown in FIG. **6**C and FIG. **6**D, the user X or user Y or other users (who's web-page/browser did not depicted) in the same user-group-**1** have entered a group online meeting after each one login to the CCDSVM from a web browser on any the user's local system across said network. Further, all users can get an identical web-page provided by the console support software **6** of the control system **2** except with different contents in the displayed private work space section of the web-page due to each user has been assigned with different resources. The web-page screen layout provided by said console support software **6** contains four sections:

1) a private user message input area (section-**1**),

2) a message displaying section for common message area **18** illustrated in FIG. **4**B that all users in the user-group-**1** (section-**2**) can access,

3) a file and/or folder display section for the group common file and folder area **19** illustrated in FIG. **4**B that all users in the same user-group-**1** (section-**3**) can access, and

4) a file and folder tree displaying and management section (section-**4**) for the folder and file tree in the private file and folder area **22** of a user's private work space illustrated in FIG. **4**C.

Initially, section-**1**, section-**2**, and section-**3** are all empty due to no one has posted any file or message yet. The console support software **6** obtains each login user's a portion of assigned folders and files for the online meeting from the private file and folder area **22** of the private work space **16** of each user which stored on said storage media as illustrated in FIG. **4**C, and further converts said file or folder information into the presentable web-format (web-page) to be displayable in said section-**4** of a web-page of the web-browser **9** on each user's local system. The choice of total four sections of the web-page layout and the order of each section on the browser

screen is irrelevant to this invention since other choices are also possible, and further, this invention does not have limitation on what web-screen layout shall be. The information of file or folder includes the system name where the file or folder resided, the named and path of the file or folder, the owner of file or folder, the time stamps, the size of the file and so on without limitation.

During the interactive online meeting, the console support software modules **6** provides each user to post or un-post files or folders to said section-**3** of displayed group's common documents and contents management section with several steps. First, for example, providing said user X to click on a targeted file "Doc **1**" **24** or folder-**1** a time in the section-**4** of the displayed file and folder management area of the private work space **16**. Second, select an operation of either post or un-post from an operation menu (see FIG. **9**A, FIG. **9**B, FIG. **10**A and FIG. **10**B). Finally click on the selected operation to submit the task of either post or un-post the selected file or folder.

Each user with permission may perform un-post operation if such user has at least a previously posted file or folder, which has already displayed in the section-**3** of the common document and content management section of the web-browser **9** for all login users of the same user-group, see FIG. **6**A or FIG. **6**B for example.

If a file is selected, said file operation menu is used and if a folder is selected, said folder operation menu is used respectively. It is also possible to only provide one menu for both file and folder operation by the console support software **6**. The operation menu either can be pop-up menu for folder as shown in FIG. **9**A or pop-up menu for file as shown in FIG. **9**B, or non-pop-up menu for folder as shown in FIG. **10**A or non-pop-up menu for file as shown in FIG. **10**B. The choice of the operation menu is irrelevant to this invention although the actual implementation of this part of invention has used pop-up menu, and as matter of the facts that there is no limitation on how the operation menu is to be designed in this invention.

In the example of FIG. **6**A, the user X can post a file named "Doc **1**" **24** to said section-**3** of the common document and contents display section displayed in each user's web-browser **9** screen. In the example of FIG. **6**B, the user Y also can post a file named "picture **1**" **25** to the same section-**3** in said web-browser **9**. In addition, another not depicted user Z posted a file of "project code". The files or folders posted could be the one physically located in the control system **2** or on any provisioned system **3** of the CCDSVM.

In addition, during the interactive online meeting, the console support software **6** of the control system **2** also provides each user to be able to post message to massive users in the same group via said section-**1** of the private user message input section by typing a message and click on a "send" button. In the example of FIG. **6**A, the user X posted a message of "Msg**1** of user X" **26**. In the example of FIG. **6**B, the user Y also posted a message of "Msg**2** of user Y" **27**. In addition, another non depicted user posted a message of "Msg**3** of user Z". As a result, all of three messages were displayed in the same said section-**2** of each user's browser **9** screen.

In a specific situation, the console support software **6** also provides permitted user to un-post a previously posted message. Just for example, as illustrated in FIG. **6**A, said console support software **6** provides the permitted user X to select a previously posted "Msg**2** of user Y" **27** from said section-**2** of the common message display section of said web-browser **9** screen of the user X, and select un-post operation from said operation menu and submit the selected un-post task.

The detailed information of how the post or un-post task to be implemented will be discussed in next two paragraphs. In addition, with this invention, the action of the "post" information such as posting information of files or folders or messages or other resources is equivalent to the action of "share" information. The action of "un-post" information is equivalent to the action of "un-share" previously posted or shared information, in other words, to "remove" the previously posted or shared information.

After a user submitting information of the post or un-post task via the web-browser **9** of a console system **1** or a client system **10** or the control system **2** to the control system **2**, the console support software **6** of the control system **2** obtains and parses the information to determine: who is the original user initiating the task; what is the type of the task; what is the type of the associated resources for task to work on; who is the targeted user; and what is the time stamp, and may also include other information without limitation. The console support software **6** further determines that if the task is to post a file or folder from said original user, the console support software **6** picks up the corresponding file or folder information that from the file and folder area **22** of private work space **16** of the original user as illustrated in FIG. **4**C and deposits such information to the file and folder area **19** of the common work space **15** of said group as illustrated in FIG. **4**B in this case.

If the task is to un-post a previously posted file or folder, which is posted from a specific user, in said common work space of said group, the said console support software **6** search for the corresponding information of file or folder, in the file folder area **19** of said common work space **15** of said group after it is found.

If the task is to post a message to users in the group, the parsed message will be stored into the private message area **21** of the original user's private work space **16** and also is copied to message area **18** of the common work space **15** of said user group by the console support software **6**.

If a task is to un-post (delete/remove) a previously posted message, the console support software **6** searches for a matching message in the common message area **18** of said common work space **15** illustrated in FIG. **4**B of said user group and remove such message entry if it is found based on the message and the original user's identification and other associated information such as time stamp, security permission and so on without limitation.

It shall be noticed that the task of un-posting a file or folder of this invention does not actually delete the file and folder from original user's private work space **16**. In addition, a lock protection mechanism is deployed whenever a message or information of a file or folder is written, stored, or deposited to either said private space **16** or said common work space **15**, or to said disk drives or other storage media. Throughout the rest of discuss, the lock protection is assume to be a default action without further mentioning whenever a write/store/deposit action take place. As matter the fact, each task of posting file, folder, and message as well as task of un-posting the previously posted file or folder, or message performed by multiple users in multiple groups with this invention can be executed by the console support software **6** instantly with memory speed.

FIG. **6**A and FIG. **6**B illustrate that after the console support software **6** of the control system **2** successfully executing the tasks for each user in each group, any user in a user-group can get a fresh view of said section-**2** of the common messages display area of the user-group-**1**, and a fresh view of said section-**3** of the common documents & contents display

area of the user-group-**1** through each user's web-browser **9** on the client system **10** or on the control system **2** or on the console system **1**.

The user can manually click on a refresh-button, which does not show in the FIG. **6**A and FIG. **6**B, to refresh the corresponding part of browser screen for either said section-**2** or section-**3**. Also said corresponding part of browser screen for said section-**2** or section-**3** can be automatically refreshed. As mater of the fact, optionally, the console support software **6** has encoded a run-time script in a web-page, which is encoded with said presentable web-format and to be displayed in the web-browser **9** screen after each user log on to the CCDSVM and start said group online meeting.

Thereafter the run-time script will automatically and periodically perform the tasks of refreshing said section-**2** and section-**3** of the web browser **9** screen. The frequency of the periodic refreshing can be adjusted by said console support software **6** of the CCDSVM depends on the needs of the meeting, which typically can range from 1 to 3 seconds and there is no limitation on this aspect. Also, with the preferred example, said run-time script provided by the WCUWE is a JavaScript, however, other type of programming scripts or methods are also possible and there is no limitation on what kind of run-time program shall be used in this invention. In addition, the console support software **6** can provide each user the choices of either automatic or manual refreshing said web-browser **9** screen.

Upon the task of refreshing the section-**2** or section-**3** in said web-browser **9** screen is performed for each user, a request of updating such screen window will be generated and sent from said web-browser **9** to the control system **2**. The console support software **6** obtains and parses each request to determine what user group, and which part of said common work space **15** of said user-group need to be updated.

Based on the information of each parsed request, the console support software **6** of the control system **2** retrieves the updated resources and objectives from said common work space **15** of said user-group, which includes recent posted files, folders, or messages from each user in the user group, and in another embodiment which includes the updated list of files and folders or messages after un-post one or more files or folders, or messages by users in the group. Finally, the console support software **6** of the control system **2** converts retrieved information of the resources and objects to the presentable web-format (web-page).

If the request is to update the group messages, the converted information is displayed in the section-**2** of the group common message display area, via the corresponding web-page in the web browser **9**. If the request is to update the group common files and folders display area, the converted information is displayed the section-**3** of said web-browser **9** screen through the web page. Therefore, after refreshing said section-**2** and section-**3** in said web-browser **9** screen, each user can have an identical view of all posted files and folders in the section-**3** of the group common document display area of said browser **9** screen, and can have an identical view of all messages in the section-**2** of group common message display area in said web-browser **9** screen, which posted by all other users in the user-group. Meanwhile said console support software **6** still allow each user kept most of resources and objects in their own private work space **16** un-exposed to other users in the group during the online interactive meeting.

For example, as illustrated in of FIG. **6**A, the user X at his/her web-browser **9** can see "picture **1**" **25** posted by the user Y in the group common file and folder display area but can not see other information in the user Y's private work space **16** area, and plus can see the "Doc **1**" posted by the user X himself. Also, as illustrated in of FIG. **6**B, the user Y at his/her web-browser **9** can see "Doc **1**" **24** posted by the user X in the group common file and folder display area but can not see other information in the user X's private file and folder display area in the user X's private workspace **16**, and plus can see the "picture **1**" posted by the user Y himself.

A difference between a conventional Internet group communication model and the group based communication model described in the present invention is that the console support software **6** of the control system **2** provides each user to control post or un-post said resources from user's own private work space while the conventional Internet communication model are not capable of performing posting/un-posting task. For example, un-posting a previous posted folder with the WCUWE of the CCDSVM does not delete the materials in user's private work space **16** while the delete operation with said existing Internet group communication model will completely delete a shared file or folder. The dynamic work space of this invention provides combined security protection for information and flexibility and efficiency for users and groups of users to have the online meeting within the WCUWE of the CCDSVM environment.

In addition, with a preferred example, since the posted information of each files are encoded by said console support software **6** with a web link and displayed in the section-**3** of the web-browser **9** screen, therefore, each user in a user-group can click on said link to download the posted file, which provided in the private file and folder area of the private work space **16** of their peer users in the user-group, from a system where file is actually located. For example, the user X can download said "picture **1**" posted by the user Y, where the "picture **1**" may reside either on the control system **2** or on any provisioned system **3**,; and the user Y can also download said "Doc **1**" posted by the user X, where the "Doc **1**" may also reside either on the control system **2** or on any provisioned system **3**. This achieves a true peer-to-peer exchanging files resided on permitted systems between permitted users in a same user group. In addition, there is no size limitation on the file size for file to be downloaded, which is a gifted benefit from Internet technology. Besides the exampled and understandable method of download, other methods of retrieve peer users' digital documents or contents are also possible by deploying a more efficient proprietary file transfer mechanism to achieve the same results of said exchanging file. It shall be noted that the said method of download is only applicable to said files and does not apply to folder and other said resource information. To obtain a targeted peer user's entire folder and the underneath files and sub folders, a special files and folder transfer method is required.

As shown in FIG. **9**A & **9**B, and FIG. **10**A & **10**B, the console support software **6** of the control system **2** also provide each user to perform many other tasks during online meeting through said operation menu such as upload, rename, delete, copy/paste and so on without limitation. The upload operation allows each user to dynamically upload files one a time through the web-browser **9** from the local storage of the client system **10** or console system **1** into said user's private work space on the control system **2** during the meeting, and further to be posted into said common work space of an user-group which each user belong to.

The mouse click driven copy/past operation allows the files and folders from the permitted folders of the permitted provisioned system **3** to be dynamically and physically copied to each users' private work spaces on the control system **2**, and the information of said files and folders could further to be displayed via said web-page in the web browser **9** screen on each user's local system and to be further posted into the

common work space **15** of the user-group that each user belongs in addition to directly posting these files or folders on the provisional systems as described in previous paragraph without physically copying over. As matter of the fact, operations provided by the console support software **6** of the control system **2** have added efficiency to posting or un-posting files and folders for each user depending on the real needs.

Additionally, the console support software **6** provides each user of each user group with capability of saving the posted messages in the common message area **18** of said common work space **15** of said each group into a file in the common file & folder area of said common work space **15** of the group. Further, the saved file can be viewed, and downloaded by each user in the group after using the method of posting. Besides whenever there is needs, the console support software modules **6** provide users to reset message area to be empty by removing all message entries in the message area of said group's common work space **15**.

The important fact is that the designed WCUWE of the CCDSVM has provided concurrent users each from a single web browser on the user's local system anywhere on the network to securely perform permitted various tasks, which could be run concurrent within a single web-browser. Because of the console support software modules **6** of the control system **2** provides each users have exclusive view for resource objects in their private work space **16** and in said common work space **15** of a user-group which each user belongs to, and provides each user to manage resource objects through their own private space, the multiple users in multiple groups can concurrently post or un-post files, folders, messages, or other resource objects without interfere users in other user-groups. Thus, the work spaces of each user and each group are well protected.

Hence, the console support software modules **6** of a control system **2** provides privileged users to perform tasks of creating multiple users and user-groups on the control system, and each group can be assigned with variable size of users for online interactive meeting. The total number of the users and the total number of the user groups, and maximum number of the users and the total number of the user-groups, and the maximum number of the users per groups that the WCUWE of the CCDSVM can support is an implementation issue and depends on the capacity of the control system **2**. The WCUWE of the CCDSVM and CCDSVM has been designed with unlimited scalability for both users and provisioned systems **3**. On the other hand, the spirit of this invention does not limit to with the boundary issues of the WCUWE and the CCDSVM.

FIG. **8**A and FIG. **8**B illustrate that each user-X and user-Y has a different private work space **16**, and each user has assigned with different system resources. Without joining any group or participating any group meeting, the console support software **6** of the control system **2** can also provide user X and user Y to have peer-to-peer model for instantly posting files or folders one a time to each other, or instantly un-post the previous posted files or folders through similar steps as the posting or un-posting materials in group online meeting model as follow:

1) provide each user able to select, via clicking on a source of file or folder for post or un-post;

2) provide the user to selecting a post operation task if user want to post any selected file or folder through operation menu; or provide user selecting un-post a previously posted file or folder;

3) upon submitting the operation task, let each user to provide input to a prompt of "who is a targeted user?" which provided by said console support software **6**, and further submit the selected operation task;

4) the console support software **6** obtains and parses task information including of the original user, the source of file or folder, type of operation, and other information;

5) If said type of operation is to post a file or folder, said console support software **6** to deposit the information of said source file or folder to said private file and folder area of said private work space **16** of a targeted user. If said operation type is un-post a file or folder, said console support software **6** search a corresponding file or folder information from said private file and folder area of said private work space **16** of said targeted user to find if it matches the source file or folder, if a match is found, the corresponding entry of the file or folder is removed from said private file and folder area of said private work space of the targeted user; and

6) Upon the target user refreshing the web-page displayed in the web-browser **9** screen, the console support software **6** retrieves and converts the updated information in said private file and folder area of the private work space **16** of the targeted user into a presentable web-format (web-page) and causes display of said web-page in the web-browser **9** screen for viewing.

For example, as shown in FIG. **8**A, the user X from a web browser **9** can follow said steps 1) to 3) to post his "folder **2**" **32** in the displayed files and folders area of the private working space **16** in the user X's web-browser **9** screen to into the user Y's user interface such as shown in FIG. **8**B; upon the user Y refreshing his/her web-browser **9** screen on the user Y's local system by walking over any system node or folder node in the displayed resource tree, the user Y will see said "folder **2**" **32** in his/her displayed private file and folder tree in the private work space **16** of the user Y such as shown in FIG. **8**C. As matter of the fact, when walking over a system node, or file & folder tree displayed in said web-browser **9** screen, a request of refreshing web-page screen is generated and to be handled as described in the steps described above.

In addition to post messages in the group based communication involving parties with two or more users, the WCUWE of the CCDSVM also provide each user to post message to each other user in non-group based 1-to-1 personal model. The FIG. **7**A and FIG. **7**B have shown an example of the user X and user Y, who may not be in a same user-group, however, the console support software **6** of the control system **2** provides each user to share and exchange messages with each other through a three-section web-page being displayed in said web browser **9** screen.

The section-**3** of the web-page in said browser **9** let each user to work with same steps of non-group based 1-to-1 posting files and folders or un-posting previously posted files or folders as described previously. The section-**3** included in FIGS. **7**A and **7**B just for a demonstration of how flexible of this invention is for displaying information in the private work space **16** of each user. Further, the number of sections in the screen layout and the order of each section in the web-page display really is not a limitation to this invention.

The section-**1** is an user message input area together with an additional input field of "To who" due to unlike in group online meeting, the console support software modules **6** have no knowledge of who is the targeted user in this case. Thus, a user can posting a message with similar steps as previously discussed previously in this invention. Nevertheless, unlike with group meeting messaging model, after obtaining and parsing information of the messages from each user, the console support software **6** of the control system **2** will deposit the message to both targeted peer user and the original user's message areas of the private work space **16**. Further, with

same steps that described previously for displaying messages in each user's browser **9** screen each user can view both own message and messages from peer user in section-**2** of the browser **9** screen.

For example, as shown in FIG. **7**A, the user X input a message of "msg**4** of user X" **30** and type "user Y" in input field of "To who", and then submit the message by clicking on "Send" button. Similar as shown in FIG. **7**B, the user Y can input a message of "Msg**3** of user Y" **31**, and type "user X" in input field of "To who", and then submit the message. Upon the refreshing a web-page in the web-browser **9** screen of each user, the user X will see the message **30** he/she sent and the message **31** from the user Y. Also the user Y will see both messages **30** and **31** at the same time.

Again, the design choice of three sections of the web-page display layout and the order of each section is just a preferred example of implementation for non-group based peer user communication. This invention does not limited by web-screen display layout, for example, a screen display only with said section-**1** and section-**2** are also possible and so on without limitation.

The WCUWE of the CCDSVM has provided a security hierarchy for members of any team to work together cross multiple locations. With said security hierarchy and said steps of posting or un-posting message, file and folder, the console support software **6** of the control system **2** can let only a team leader with capability of posting folder or file to be viewed, shared, by all team members or by an individual member in a non-group meeting based environment. In addition, the console support software **6** also provides each team member to post or un-post his/her files or folders to only a specific peer user one a time in non-group-meeting based environment. Again the said files or folders could be residing in said control system **2** or in any provisioned system **3**.

It is relatively straightforward for the WCUWE of the CCDSVM, more specifically the console support software modules **6**, to support massive online users, who do not have account with the CCDSVM and can not login the CCDSVM, yet to view the dynamically posted files, folders, or messages posted by the users with user account registered with the CCDSVM.

In a preferred example, such as shown in FIG. **6**E, the console support software **6** of the control system **2** provides a web-page to massive online users, who do not have to login the CCDSVM. The web-page could be layout with two major sections with section-**1** for display of messages in the sharable message area of a group's common work space **15** while the section-**2** for display of files or folders in the sharable file or folder area in said group's common work space **15**.

For users belong to same said group, each user from a web-browser **9** screen displayed on the user's local system is able across said network to login to the CCDSVM and obtain the group online meeting web-page as shown in FIG. **6**A for the user X and in FIG. **6**B for the user Y. Further as described previously, each user in said user group can instantly post message to said message area **18** of said group's common work space **15**, or post file or folder to the sharable file and folder area in the group's common work space **15**. Also, each user in the group can instantly un-post the previously posted files or folders by removing the corresponding file or folder entry in the file and folder area of the group's common work space **15**, or un-post the previously posted messages from the common message area of the common work space **15** of said group as described in previous.

Meanwhile as previously described and as shown in FIG. **6**E, said web-page for said massive online users, who have no account with CCDSVM, can be dynamically updated, displayed, and viewed by said massive users anywhere through their own web browser accordingly with the same pace as dynamically displaying in said section-**2** and section-**3** via a web-page in the web browser **9** screen as shown in FIG. **6**A and FIG. **6**B. Again, said two sections of the web-page display layout for said massive users is a preferred example and it does not limit this invention because with this invention other layout is also possible, for example, it may be desired to have web-page layout only contains one section, which is either for messages or file or folder's display depending on the application's need.

As shown in FIG. **8**A, FIG. **8**B, and FIG. **8**C, with non-group based peer-to-peer post or un-post files and folders model, people may often do not remember how many files and folders have been posted for sharing. For example, as shown in FIG. **8**A, the user X can select "folder **2**" **32**, then select a post operation from an operation menu, and provide a target "user Y" to share the "folder **2**" **32** with the user Y. As show in FIG. **8**C, the user Y can see "Folder **2**" **32** posted by the user X in the web-browser **9** screen of the user Y after the user X posted "Folder **2**" **32**. If the user X selects and posts various files or folders to other dozen people for share, the user X will have difficult time to remember what has been shared and who are targeted users. As shown in FIG. **11**, the console support software modules **6** of the control system **2** can provide each user, the user X for example, a summarized view of a sharing list of how many files or folders being posted by the user X and shared with which user.

As shown in FIG. **11** for the personal share management, in the user X's web-browser **9** screen, an entry of posted "Folder **2**" **32** is listed at bottom of said sharing list and it is posted to and shared with the user Y. Further, said console support software **6** can provide each user selectively un-post the previously posted files and folders by clicking on the "delete shared?" of an operation menu for a corresponding entries in the displayed share list and further to generate an un-post operation task. For example, the user X clicks on "delete share?" to un-post "Folder **2**" **32**. Thereafter, as described before said console support software **6** of the control system **2** will execute the un-post task. Further upon refreshing the targeted user Y's web-page in the web-browser **9**, in stead of seeing "Folder **2**" **32** as shown in FIG. **8**C, the user Y will see an updated display of folders via an updated web-page as shown in FIG. **8**B, where the "Folder **2**" **32** has been removed. The operation menu for "delete share?" depicted here just for example and it could be displayed in any other form.

In addition to let original user in a user group to control the un-posting file or folder in the group based communication, the console support software **6** of the control system**2**) may also provide a privileged user to un-post the previously posted files or files in said displayed file and folder section of a web-page. To support this capability, the console support software **6** will provide an operation menu to associate with said displayed files and folders in said display section, further to allow privileged user to perform similar un-post operation. After receiving the un-post operation, the console support software **6** will perform similar said searching for selected file or folder in the group file and folder area **19** of the group common work space **15** of said group, if there is a match, the corresponding entry will be deleted. Also, the file and folder section of the web-page in the web-browser **9** can be refreshed accordingly as described before.

Having briefly described embodiments of the web based network environment in which the present invention operates, FIG. **12** illustrates an example of a computer system **200**, which is an exemplary client system **10**, provisioned system **3**, control system **2**, or console system **1** in which the features

of the present invention may be implemented. Process of the web based information exchange can be implemented in any processor-based computer system, such as a PC, a workstation, or a mainframe computer. It will be apparent to those of ordinary skill in the art that other alternative computer system architectures may also be employed.

Referring back to FIG. **12**, computer system **200** includes a processing unit **201**, an interface bus **211**, and an input/output ("IO") unit **220**. Processing unit **201** includes a processor **202**, a main memory **204**, a system bus **211**, a static memory device **206**, a bus control unit **205**, a mass storage memory **207**, and a web information exchange module **230**. Bus **211** is used to transmit information between various components and processor **202** for data processing. Processor **202** may be any of a wide variety of general-purpose processors or microprocessors such as Pentium™ microprocessor, Motorola™ 68040, or Power PC™ microprocessor.

Main memory **204**, which may include multiple levels of cache memories, stores frequently used data and instructions. Main memory **204** may be RAM (random access memory), MRAM (magnetic RAM), or flash memory. Static memory **206** may be a ROM (read-only memory), which is coupled to bus **211**, for storing static information and/or instructions. Bus control unit **205** is coupled to buses **211**-**212** and controls which component, such as main memory **204** or processor **202**, can use the bus. Bus control unit **205** manages the communications between bus **211** and bus **212**. Mass storage memory **207**, which may be a magnetic disk, an optical disk, hard disk drive, floppy disk, CD-ROM, and/or flash memories for storing large amounts of data. The web based information exchange **230**, in one embodiment, is a network control component for facilitating information exchange over the Web. It should be noted that web based information exchange **230** could be software or hardware or a combination of software and hardware components.

I/O unit **220**, in one embodiment, includes a display **221**, keyboard **222**, cursor control device **223**, and communication device **225**. Display device **221** may be a liquid crystal device, cathode ray tube ("CRT"), touch-screen display, or other suitable display device. Display **221** projects or displays images of a graphical planning board. Keyboard **222** may be a conventional alphanumeric input device for communicating information between computer system **200** and computer operator(s). Another type of user input device is cursor control device **223**, such as a conventional mouse, touch mouse, trackball, or other type of cursor for communicating information between system **200** and user(s).

Communication device **225** is coupled to bus **211** for accessing information from remote computers or servers, such as server or other computers as illustrated in FIG. **1**, through network **11** or **12** as illustrated in FIG. **1**. Communication device **225** may include a modem or a network interface device, or other similar devices that facilitate communication between computer **200** and the network. Computer system **200** may be coupled to a number of servers via a network infrastructure such as the infrastructure illustrated in FIG. **1**.

The present invention has been described in considerable details with reference to certain examples. However, other versions and examples are also possible, therefore, the spirit of this invention shall not be limited to these examples and/or embodiments. Also, the claims of this invention will label with numbering and in most cases, the numbering does not represent any sequence or order unless specifically described as the steps of a method.

The invention claimed is:

1. A method for supporting virtual presentation between a plurality of users in a collaboration system, the method comprising:

displaying a first user interface comprising metadata of files and folders, residing in a server or in at least one computing device, on a first end-user device to allow a first user to select one selected file or one selected folder from the metadata displayed and request metadata of the selected file or folder to be posted to a second interface;

causing the server to store the metadata information, but not content, of the selected file or the selected folder according to the request for the posting received from the first end-user device; and

displaying to a second user the stored metadata of the selected file or the selected folder including to display a graphic indicator of the selected file or folder in the second user interface on a second end-user device to allow the second users access to the content of the selected file or folder through the stored metadata displayed in the second end-user interface.

2. The method as recited in claim **1**, wherein the stored metadata of the selected file or folder is stored in a file and folder section in a workspace configured in a storage medium, the workspace accessible by the second user and comprising a plurality of section.

3. The method as recited in claim **1**, wherein the metadata of the files and folders at least comprises name, path, owner, or timestamps of each of the files and folders.

4. The method as recited in claim **1** further comprising:

configuring a message section in the first user interface to allow the first user to input a message thereto and request the message to be posted to the second user interface;

causing the server to store the message according to the request for posting the message received from the first end-user device; and

displaying to the second user the stored message in the second user interface on the second end-user device for the second user access to the message.

5. The method as recited in claim **4**, wherein the message is stored in a message section in a workspace configured in a storage medium accessible to the second user, the message including predetermined information in addition to content of the message.

6. The method as recited in claim **4**, further comprising:

allowing the first user via the first user interface to request removing the displayed message; or information of the selected file or folder from the second user interface;

causing the server to delete the stored message or metadata information of the selected file or folder according to the request for the removing received from the first end-user device; and

displaying to the second user the second user interface, without including the deleted previously stored message or information of the selected file or folder, on the second end-user device.

7. The method as recited in claim **1**, further comprising:

wherein the first end-user device and the computing device are the same device or different devices separate across a network;

wherein the second user device and the computing device are the same device or different devices separate across a network.

8. The method as recited in claim **1**, further comprising:

wherein the first end-user device and the server are the same device or different devices separate across a network;

wherein the second end-user device and the server are the same device or different device separate across a network.

9. A server in a collaboration system supporting virtual presentation between a plurality of users, the server comprising:

at least one hardware processor, and

program code which, when executed by the at least one hardware processor, causes the server to:

display a first user interface comprising metadata of files and folders, residing in the server or in at least one computing device, on a first end-user device to allow a first user selecting one selected file or one selected folder from the metadata displayed and requesting the metadata of the selected file or folder to be posted to a second user interface;

store the metadata information, but not content, of the selected file or the selected folder according to the request for the posting received from the first end-user device; and

display to a second user the stored metadata of the selected file or the selected folder including to display a graphic indicator of the selected file or folder in the second user interface on a second end-user device to allow the second user access to the content of the selected file or selected folder through the stored metadata displayed in the second user interface.

10. The server as recited in claim **9**, wherein the stored metadata of the selected file or the selected folder is stored in a file and folder section in a work space configured in a storage medium, the workspace accessible by the second user and comprising a plurality of sections.

11. The server as recited in claim **9**, wherein the metadata of the files and folders are collected by the server locally and/or collected automatically from the computing device across a network which is one of a corporate intranet, the Internet, a local area network or a wide area network.

12. The server as recited in claim **9**, wherein the program code causes the server further to:

configure a message section in the first user interface to allow the first user to input a message thereto and request the message to be posted on the second user interface;

store the message according to the request for posting the message received from the first end-user device; and

display to the second user the stored message in the second user interface on the second end-user device for the second user access to the message.

13. The server as recited in claim **12**, wherein the program code further configures the server to:

allow the first user via the first user interface to request removing the displayed message or information of the selected file or the selected folder from the second user interface;

delete the stored message or the stored metadata of the selected file or the selected folder according to the request for the removing received from the first end-user device; and

display to the second user the second user interface, without including the deleted previously stored message or metadata of the selected file or folder, on the second end-user device.

14. The server as recited in claim **12**, wherein the message is stored in a message section in a workspace configured in a storage medium and accessible to the second user, the message including predetermined information in addition to content of the message.

15. The server as recited in claim **9**, wherein the metadata of the files and folders at least comprises name, path, owner, or timestamp of each of the files and folders.

16. The server as recited in claim **9**, wherein the program code further causes the server to:

allow the first user to present a message, file information or folder information to a third user on a third end-user computing device; and

allow each of the second and third users to present a message, file information or folder information to the first user.

17. A collaboration system supporting virtual presentation between a plurality of users, the collaboration system comprising:

a server; and

at least one computing device coupled to the server across a network,

wherein the server is configured to:

display a first user interface comprising metadata of files and folders, residing in the server or in the computing device, on a first end-user device to allow a first user selecting one selected file or one selected folder from the metadata displayed and request metadata of the selected file or folder to be posted to a second user interface;

store the metadata, but not content, of the selected file or the selected folder according to the request for the posting received from the first end-user device; and

display to a second user the stored metadata of the selected file or the selected folder in the second user interface on a second end-user device to allow a second user access to the content of the selected file or folder via the stored metadata displayed in the second user interface.

18. The collaboration system as recited in claim **17**, wherein said network comprises one of a corporate intranet, an internet, a wide area network, or a local area network.

19. The collaboration system as recited in claim **17**, wherein each of the computing device and the server is operable to execute and respond to a request for access to the content of the selected file or folder, wherein the content of the selected folder contains at least one first file, or one first subfolder, the first subfolder may further comprise at least one second file or one second subfolder.

20. The collaboration system as recited in claim **17**, wherein each of the first and second user interfaces is dispalble via execution of a web browser, wherein said displaying metadata information of a file or folder comprises displaying a graphic indicator to represent the file or folder.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 8,713,442 B2
APPLICATION NO. : 13/080007
DATED : April 29, 2014
INVENTOR(S) : Sheng Tai Ted Tsao

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

IN THE CLAIMS:

Two occurrences in col. 22, line 8, Claim 1; two occurrences in col. 23, line 13, Claim 9; and two occurrences in col. 24, line 31, Claim 17, please replace “one selected” with ---one---;

In col. 22, line 11, Claim 1, and lines 49-50, Claim 6; in col. 23, line 17, Claim 9; and in col. 24, line 55, Claim 20, please replace “metadata information” with ---metadata---;

In col. 22, line 47 and line 55, Claim 6; and in col. 23, line 51, Claim 13, please replace “information” with ---metadata---;

In col. 23, line 10 and line 21, Claim 9; in col. 24, line 1, Claim 13, line 28 and line 37, Claim 17; in col. 23, line 45, Claim 12; and in col. 24, line 1, Claim 13, please replace “display” with ---cause displaying---;

In col. 23, line 8, Claim 9, please replace “program code” with ---program code in a storage medium---;

In col. 24, line 49, Claim 19, please replace “contains” with ---comprises---;

In col. 24, line 50, Claim 19, please replace “may” with ---operable to---;

In col. 24, lines 53-54, Claim 20, please replace “dispalble” with ---operable to be displayed in---.

Signed and Sealed this
First Day of July, 2014

Michelle K. Lee

Michelle K. Lee
Deputy Director of the United States Patent and Trademark Office

Exhibit D

US008799473B2

(12) **United States Patent**
Tsao

(10) **Patent No.: US 8,799,473 B2**
(45) **Date of Patent: *Aug. 5, 2014**

(54) **CONCURRENT WEB BASED MULTI-TASK SUPPORT FOR COMPUTER SYSTEM**

(75) Inventor: **Sheng Tai (Ted) Tsao**, San Jose, CA (US)

(73) Assignee: **Sheng Tai (Ted) Tsao**, Fremont, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1195 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **12/075,314**

(22) Filed: **Mar. 4, 2008**

(65) **Prior Publication Data**

US 2008/0178184 A1 Jul. 24, 2008

Related U.S. Application Data

(63) Continuation of application No. 10/713,904, filed on Aug. 6, 2002, now Pat. No. 7,418,702.

(51) **Int. Cl.**
G06F 15/173 (2006.01)

(52) **U.S. Cl.**
USPC **709/226**; 709/229; 718/100

(58) **Field of Classification Search**
USPC 718/1, 100; 709/202, 203, 217, 219, 709/226, 229
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,769,019	B2 *	7/2004	Ferguson	709/219
7,165,256	B2 *	1/2007	Boudnik et al.	718/104
7,188,343	B2 *	3/2007	Sanchez et al.	718/100
2002/0178297	A1 *	11/2002	Lister et al.	709/310
2004/0030643	A1 *	2/2004	Madison et al.	705/39

* cited by examiner

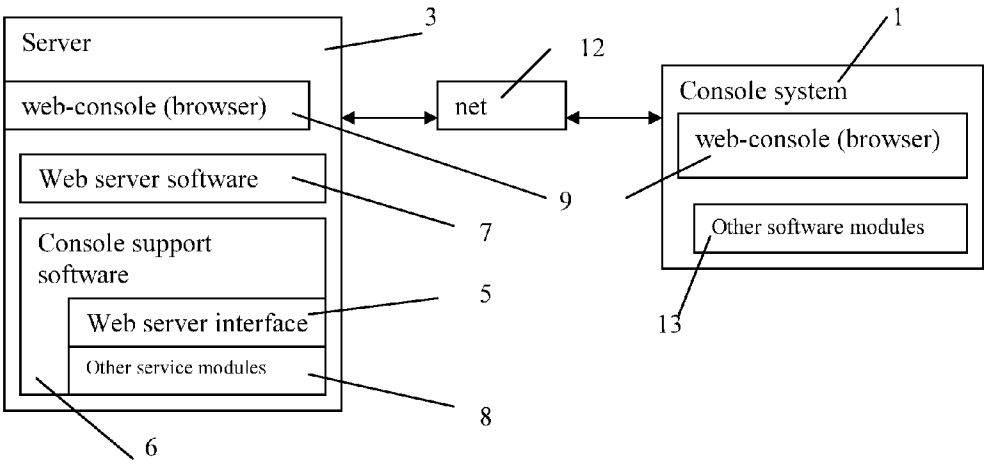
Primary Examiner — Mohamed Ibrahim

(57) **ABSTRACT**

The traditional web based computing system does not support web based multitasking. Therefore, when a user via a web user interface displayed in a web browser on an end-user computing device to submit a task of accessing a web server, the user has to wait the task to be finished before can submit a second task from the same web user interface. A method of supporting web based multitasking is disclosed herein for solving above mentioned issue to let user much efficiently perform web based tasks without wasting waiting time.

5 Claims, 7 Drawing Sheets

Console support in a Simple Environment



Console support in a Simple Environment

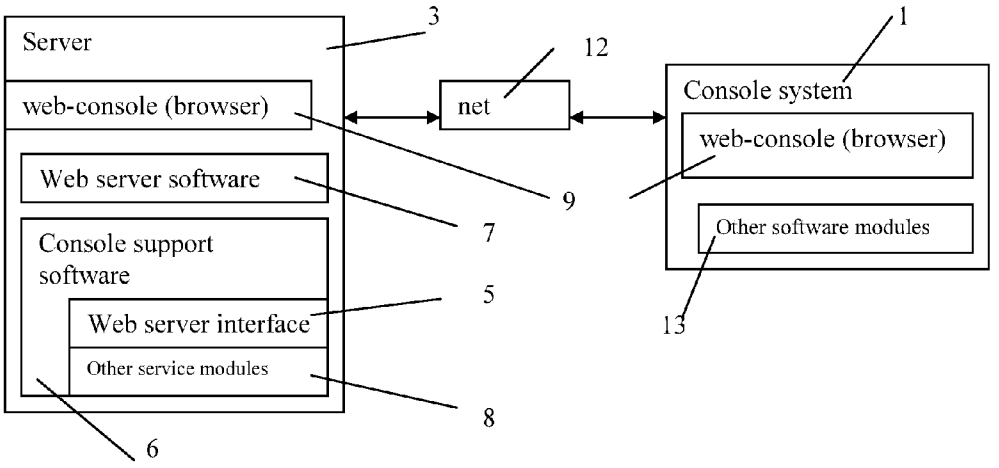


Fig. 1

Console support in a CCDSVM environment

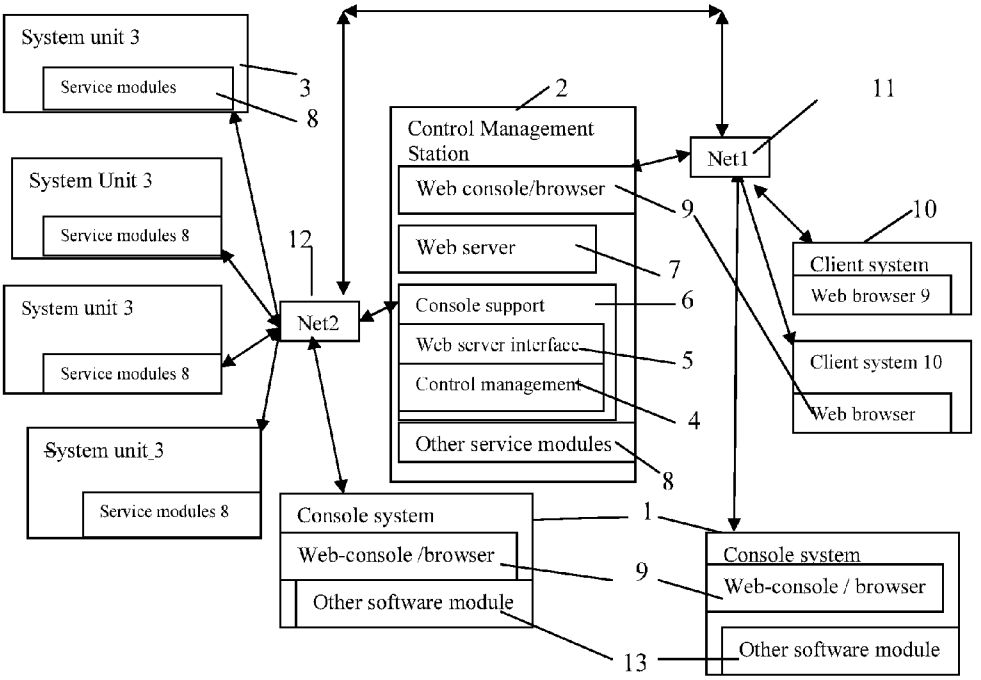
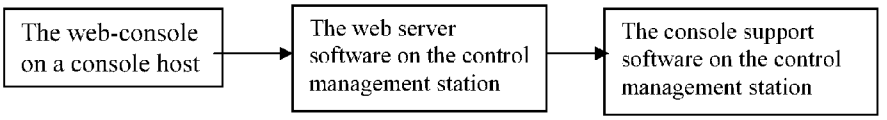


Fig. 2

Data traveling between a web-console and the console supporting software

a) Data Path 1:
Referred as sending data from the web-console-to the console support software or referred as the console supporting software receiving data from the web-console.



b) Data Path 2: (reverse path)
Referred as sending data from the console supporting software to the web-console or referred as web-console receiving data from the console supporting software.

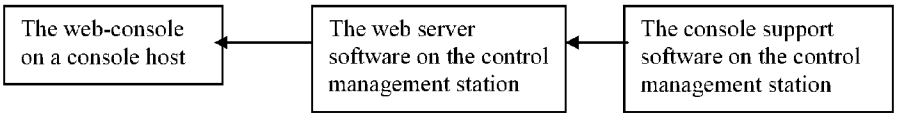


Fig. 3

Basic Task & Operation Processing Flow Chart in a CCDSVM environment.

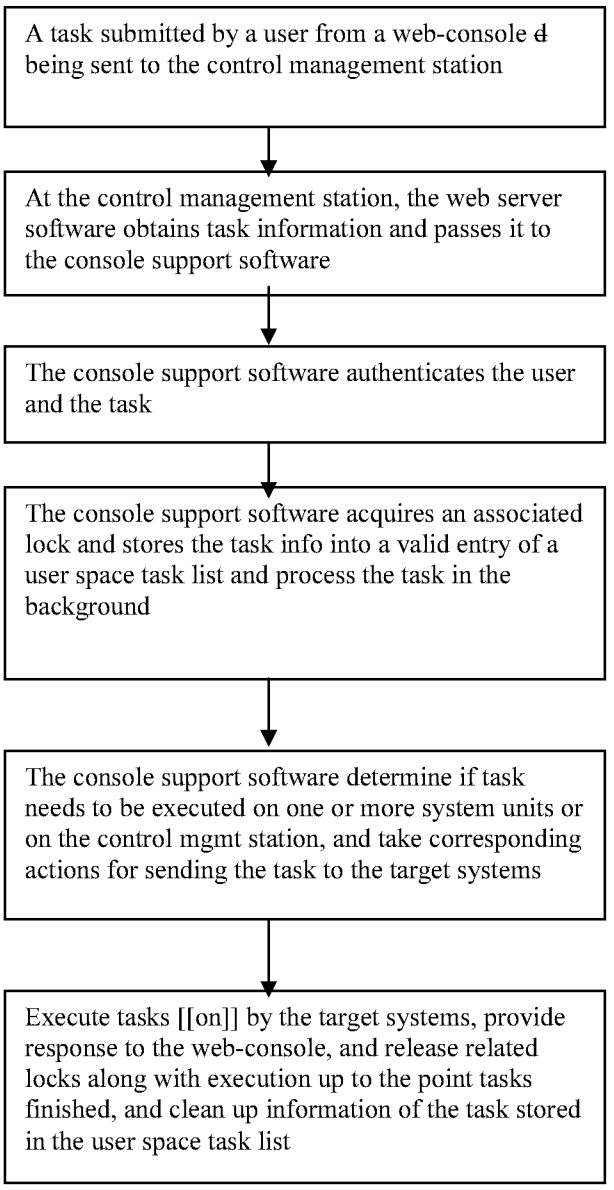


Fig. 4

The abstract data structure for supporting multiple concurrent tasks and operations in a CCDSVM environment.

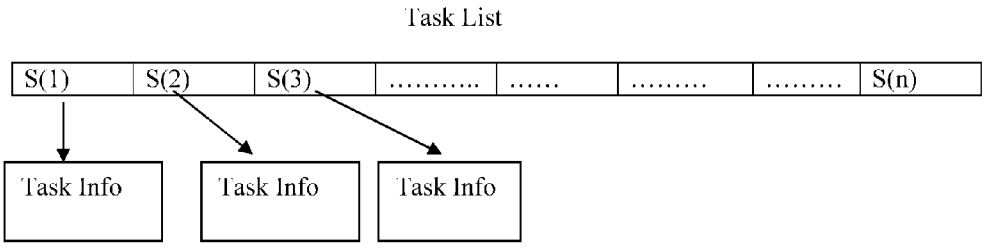


Fig. 5

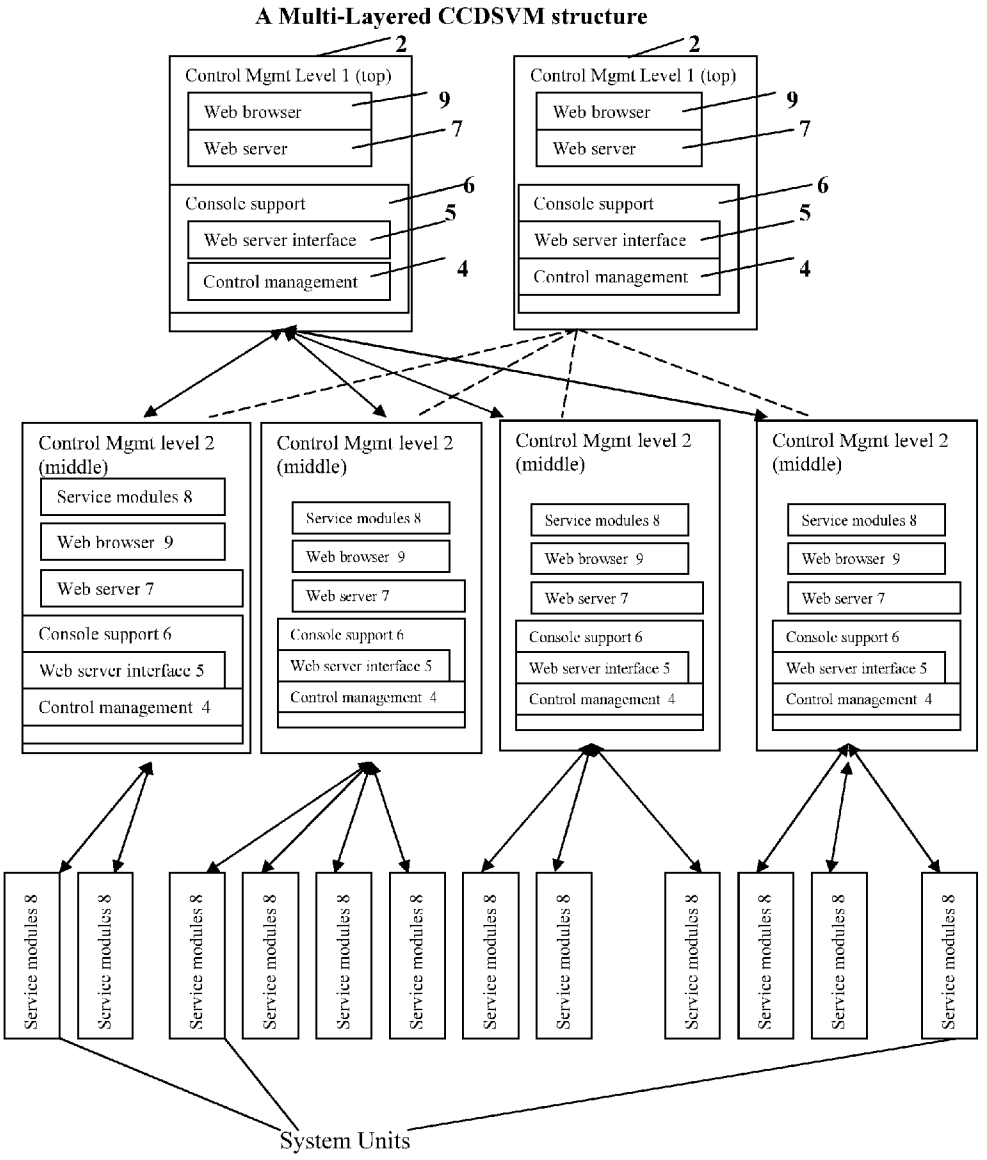

A Multi-Layered CCDSVM structure
2
Control Mgmt Level 1 (top)
Web browser
9
Web server
7
Console support
6
Web server interface
5
Control management
4
2
Control Mgmt Level 1 (top)
Web browser
9
Web server
7
Console support
6
Web server interface
5
Control management
4
Control Mgmt level 2 (middle)
Service modules 8
Web browser 9
Web server 7
Console support 6
Web server interface 5
Control management 4
Control Mgmt level 2 (middle)
Service modules 8
Web browser 9
Web server 7
Console support 6
Web server interface 5
Control management 4
Control Mgmt level 2 (middle)
Service modules 8
Web browser 9
Web server 7
Console support 6
Web server interface 5
Control management 4
Control Mgmt level 2 (middle)
Service modules 8
Web browser 9
Web server 7
Console support 6
Web server interface 5
Control management 4
Service modules 8
Service modules 8
Service modules 8
Service modules 8
Service modules 8
Service modules 8
Service modules 8
Service modules 8
Service modules 8
Service modules 8
Service modules 8
System Units

Fig. 6

The typical hardware components of a computing device such as a control management station, a system units, and a console host.

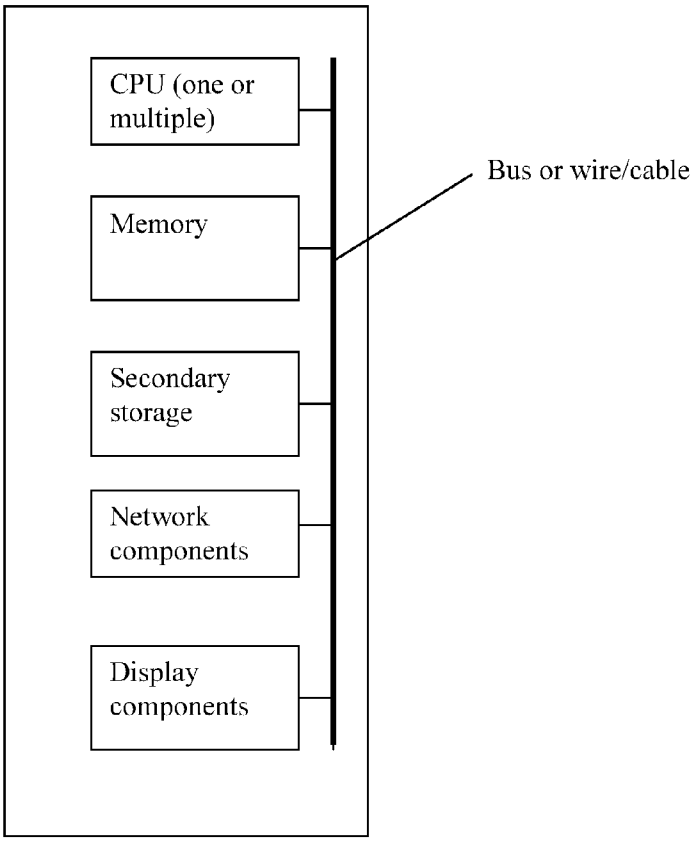


Fig. 7

CONCURRENT WEB BASED MULTI-TASK SUPPORT FOR COMPUTER SYSTEM

CROSS-REFERENCES TO RELATED APPLICATION(S)

This application is a continuation application of (a) U.S. patent application Ser. No. 12/075,314 filed on Mar. 4, 2008, which itself in turn is a continuation application of U.S. patent application Ser. No. 10/713,904 filed on Aug. 6, 2002 and is converted from provisional application 60/401,238, and now is a U.S. Pat. No. 7,418,702. All above applications are herein incorporated by reference in their entireties for all purpose.

FIELD OF THE INVENTION

The present invention generally relates to support web based multitasking for a single computing device or for a central controlled distributed scalable virtual machine system ("CCDSVM") with respect to a web based computer user work environment.

BACKGROUND OF THE INVENTION

A typical computer system provides a computer user work environment to end users, wherein the computer user work environment runs on top of a generic computer operating system. With this work environment, an end user can login to the computer system and access to various computer resources based on his or her permitted role. Therefore, the end user, for example, can perform tasks such as configure computer resources of disk drives, networks, files, folders, and directories on the computer system, and access to various computer application, and the computer operating system of the computer system provides the results of the tasks to the end user. Specially, with a generic computer operating system, this computer user work environment allows multiple concurrent users to perform multiple concurrent tasks or operations.

The computer user work environment has evolved from paper tape & punch card environment, to command line environment, and further to window environment for a local computer system in the past. This invention provides users a web-based computer user work (operating) environment on top of generic operating system for a single or multiple computers, and allows each of the users access to one or multiple computing systems through a conventional web-browser.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated into and constitute a part of this specification, illustrate one or more examples of embodiments and, together with the description of example embodiments, serve to explain the principles and implementations of the embodiments.

In the drawings:

FIG. **1** illustrates an example of a simplified web based multi-tasks support in a simple environment.

FIG. **2** illustrates an example of a simplified web based multi-tasks support in a CCDSVM environment.

FIG. **3** illustrates an example of basic data flow between a web-console on a console host and the console supporting software in a control management environment.

FIG. **4** illustrates an example of operation flow for processing a user submitted task in the CCDSVM environment.

FIG. **5** illustrates an example of a user space task list, which represents an abstraction of data structure, for controlling multiple simultaneous concurrent tasks and operations in the CCDSVM environment.

FIG. **6** illustrates one embodiment of a multi-layered CCDSVM structure.

FIG. **7** illustrates an embodiment of typical hardware components for a computing device.

BRIEF DESCRIPTION OF THIS INVENTION

The traditional web server or other server (**3** of FIG. **1**) may support a user from a web browser (**8** of FIG. **1**) displayed on a computing device somewhere on the network to perform tasks of access to the server, where the task could get quick response and could be finished in a short period of time, for example, checking a web server's status or getting the server's other information, etcetera. However, the traditional server or web server) does not support for multiple concurrent tasks or operations submitted from the same web-browser, and this problem will be escalated and manifested to people, especially when these tasks are time consuming to be finished.

For example, creating a 60 GB file system on the server, or configuring a raid controller on that server **3** illustrated in FIG. **1** is such a time consuming task. Because these tasks often take a large amount of time to be finished, and thus the user will experience the task being hang and blocking in the web-console (web browser) window on a console host **1** illustrated in FIG. **1**. As a result, no other tasks could be performed in parallel from the same web-console at the same time. In addition, traditional console supporting software does not work for a more complicated environment such as the CCDSVM environment illustrated in FIG. **2**.

To solve these problems and effectively to support multiple simultaneous concurrent tasks in a web-console for both the simple computing environment illustrated in FIG. **1** and the CCDSVM environment, the console supporting software **6** in FIG. **1** needs to include additional control management software modules **5** illustrated in FIG. **2** and others. As illustrated in FIG. **2**, the control management software module **4** shall communicate and control all system units **3** and each of the system units **3** needs service software modules **8** to communicate with control management software **4** of console support software **6**.

In addition, a user space task list illustrated in FIG. **5** could be used together with conventional or non-conventional locks to support all multiple simultaneous concurrent tasks and operations. With this invention, the multi-tasks support for the web-console in a simple computing environment shown in FIG. **1** has been viewed as a special case of such support in a CCDSVM environment shown in FIG. **2**. The CCDSVM will be degenerated into a simple server) if there is no multiple system units **3** that are illustrated in FIG. **2**.

These and other features, aspects and advantages of the present invention will become understood with reference to the following description, appended claims and accompanying figures set forth below.

DETAILED DESCRIPTION OF THE INVENTION

The CCDSVM in a embodiment is configured to provide a control management station ("control system") to control a group of computing systems and provide distributed services to at least one client system across Intranet, the Internet as well as a LAN environment. The software components of the CCDSVM form a virtual operating environment.

When a server provides software support to allow a user, via a web-browser on a computing system (device) such as a desktop, laptop, server, PDA, or cell phone, access to and manage the server, this web-browser is often referred as a web-console. With the CCDSVM, a permitted user from the web-console should be able to access and operate the entire CCDSVM.

To simplify the discussion, the term of thread and process are roughly used without differentiation between them in this invention regardless of the very restricted definition of the thread and process in the field of computer science. Here both thread and process are basically referred as a sequence of instructions based on a piece of program code that starts to be executed by a computer system step by step to carry out a computer task.

Lock is a mechanism that allows a thread to look a computer resource for its own use and prevents other threads from access to the same computer resource at the same time. There is conventional lock which can be acquired and released by the same thread. The conventional lock mechanisms have used by most software developer crossing the software industry. The lock described in this invention may or may not be a conventional one. The non-conventional lock mechanisms created in this invention can be acquired by one thread and may be released by same thread or by another thread. Therefore, it is non-conventional lock serving threads on the computing system.

FIG. **1** illustrates an example of a web-console scheme in a simple computing environment that includes a console host **1**, a server **3**, and a communication network **12**. The console host **1** further includes a web-console **9** (a browser). The server **3** includes a native web-console **9**, a web server software **7**, and a console support software **6**, where the console support software **6** further includes web server interface **5** and other service modules **8**. It should be noted that the same reference indicators will be used throughout the drawings and the following description to refer to the same or like items.

The console host **1**, from which a user is able to perform system tasks or operations for the server **3** through the web-console (browser) **9**. The console host **1** can be any computing system on the network **12** such as a server, a desktop PC, a laptop PC, a hand held PDA, or a cell phone. The web browser **9** may be commercially available software from any vendor or a proprietary software. The web browser **9** is able to handle web protocol such as HTTP. The console host **1** may also include other software modules **13** that may be implemented with any suitable programming languages such as C, C++, Java, XML, et cetera. The other software modules **13** are used to communicate between the server **3** and the console host **1** using IP, non-IP or any suitable protocols for receiving and/or sending data between the console host **1** and the server **3**.

The server **3** could be a web server or any kind of computing system with web server software that includes web server software **7** and console supporting software **6**. The console supporting software **6** includes web server interface **5** and other services software modules **8**, where the other services software **8** is operated natively on the server **3**. The web server software **7** may be a commercially available software or proprietary software, which is able to accept and handle the web protocol such as HTTP. A native web-console (browser) **9** enables a user to access and operate the server computer **3** locally.

Net **12** represents a network infrastructure such as Internet, intranet, and (LAN). The net **12** includes all kind of related network equipment and media such as switches/routers, and different kind of connecting cables and wireless communication media.

FIG. **2** shows an example of a simplified block diagram for an embodiment of the CCDSVM. The CCDSVM system includes console hosts **1**, a control management station **2**, system units **3**, networks of net1 **11** and a net2 **12**, and client systems **10**. The console hosts **1** could be any computing system on the network such as a server, a desktop PC, a laptop PC, a hand held PDA, or a cell phone. A web browser **9** of the console host **1** can be used access to and operate the entire CCDSVM. The web browser **9** may be commercially available software from any vendor or proprietary software company. The web browser **9** is able to handle web protocol such as HTTP. A difference between a web-console **9** and a web browser is that the web-console **9** allows a user access to system information and performing system operation over computing systems such as in the CCDSVM environment.

The console host **1** may also include other software modules **13**, which may be implemented with any suitable programming languages such as C, C++, Java, XML etc. These software modules **13** may be used to facilitate communications between the control management station **2** and the console hosts **1** using IP, non-IP or any suitable protocols for receiving or sending data between the console host **1** and the control management station **2**. To support a non-web-based networked console, the software configured for the web console **9** shall be capable of handling protocols other than web protocols of HTTP etc. for communicating with the console supporting software **6** on the control management station **2**.

The control management station **2** could be any computing system on the network such as a server, a desktop PC, a laptop PC, or others communication device. The control management station **2** includes web server software **7** and console supporting software **6**. The console supporting software **6** includes web server interface software modules **5** and control management software modules **4**. It should be noted that the console supporting software **6** may includes others service software modules **8**. It may also have a native web browser used as a native web-console **9**. The web server software **7** discussed earlier could be commercially available software from a major vendor or other proprietary software that is able to accept and handle the web protocol such as HTTP. The web server software **7** sends data to and receives data from the web-console **9** of the console hosts **1**.

The console supporting software **6** can be implemented with any suitable languages such as C, C++, Java, XML, etc. or even implemented by using a combination of different languages as long as it provides the features and functionality described in this invention. That means it is language independent. In addition, the communication protocol used between the console support software **6** and the service software modules **8** of the system units **3** could be any suitable protocol such IP based, or non-IP based or other protocols.

There may be several fixed threads being created based on the control management software modules **4**. There are may be various number of threads that are created based on the web server interface software modules **5** for supporting each of tasks submitted by a user through the web-console **9**. All of these threads may be communicated with each other through inter-process communication and are simply referred as the thread of the console supporting software **6**. However, to simplify the discussion, they may be just referred as the console supporting software **6** without mentioning the thread at all.

If there is a need to support a less effective non-web-based networked console, there is no need for the web server software **7** and web server interfacing software module **5** being employed. Instead, an additional network software module is required that could be implemented with any suitable pro-

gramming language and any suitable communication protocol other than web protocol (HTTP). This network software module can communicate with networked console software on the console host **1** across a communication network and can communicate with the rest of the console supporting software **6** via inter-process communication mechanism.

The system unit **3** could be any computing system on the network such as a server, a desktop PC, a laptop PC, a hand held PDA, a cell phone, and any operational system. The server could be a video server, a web server, a storage block data server (SAN unit), a video monitoring device, and so forth without limits. The system unit **3** contains service software modules **8** that are capable of communicating with the outside world. For example, the service modules **8** is used to communicate with the control management software **4** of the control management station **2** for carrying out the tasks distributed from the control management station **2**, or to communicate with the clients **10** of the CCDSVM for delivering services to them, or to communicate with another system unit **3** for transferring the data. The service software modules **8** could be implemented with any suitable programming languages such as C, C++, Java, or others. It should be noted that the communication protocol could be any suitable protocol such as IP (Internet Protocol) base or other non-IP based protocol.

The net1 **11** represents any kind of communication links between the control management station **2** and the web-console **9** or the client hosts **10**. The net1 **11** could be an infrastructure of internet, intranet, LAN or others that comprises connection media such as cables of Ethernet, optical Fiber, and/or other, wireless media, bus, and includes communication equipment such as switches, routers, and/or adapters.

The net2 **12** also represents a communication infrastructure comprising communication media and equipment that are similar to the net1 **11** has, except for providing communication between the control management station **2** and the system units **3** or the web-consoles **9** across the infrastructure of internet, intranet, LAN, WAN, or other.

The client systems **10** are not part of the CCDSVM but they may requests services from the CCDSVM as shown in FIG. **2**. The role of the client systems and the connections between the client system **10** and system unit **3** will not be described in this invention since they are irrelevant to this invention.

FIG. **3** illustrates a simplified data flow between the web-console **9** on the console host **1** of FIG. **2** and the console supporting software **6** on the control management station **2** of FIG. **2**. Data travel from the web-console **9** to the console supporting software **6** includes two steps. First, the data goes from the web-console **9** to the web server software **7** of FIG. **2** via the net **11** or **12** of FIG. **2**. Second, the console supporting software **6** obtains the data from the web server software **7** of FIG. **2** via inter-process communication. To simplify the discussion of this invention, this data traveling path will simply refer to as the console supporting software **6** obtains the data from the web-console **9** or refer to as the data being sent from the web-console **9** to the console supporting software **6**.

Data traveling from the console supporting software **6** to the web-console **9** includes two reverse steps. First, the web server software **7** gets the data from the console supporting software **6** via inter-process communication. Second, the web server software **7** sends data to the web-console **9** via the net **11** or **12**. To simplify the rest of discussion, this reverse data traveling will refer to as data being sent from the console supporting software **6** to the web-console **9** or refer to as the web-console **9** gets data from the console supporting software **6**. In addition, the terms of data may also be referred as information, or information on a web page and they will be used interchangeably herein.

FIG. **4** shows the basic tasks and operation processing flow chart, which initiated from the web-console **9**.

FIG. **5** shows one embodiment of a user space task list. Each entry on the user space task list can be used to store information of a task issued from the web-console **9**. The stored information of the task in the entry will be deleted upon the task execution is completed.

FIG. **6** illustrates another embodiment of a layered CCDSVM environment, which provides a flexible scalability mechanism to efficiently support thousands of heterogeneous system units **3**. With this structure, a control management station **2** at middle layer **2** becomes a system unit **3** and is controlled by a control management station **2** at up layer **1**.

FIG. **7** illustrates a embodiment of typical hardware components for a computing device such as for the control management system **2**, system units **3**, and console hosts **1**. The typical hardware components comprises of one or more CPU, memory, secondary storage such as disk drives and/or memory sticks, the network interface cards, and display components such as monitor or others. These components are connected internally through buses.

The detailed explanation of FIG. **2** will demonstrate how multiple concurrent tasks can be initiated from a web-console **9** and can be executed either on any one of the system units **3** or on the control management station **2** according to this invention.

In one example, a user A at a web-console **9** receives an authentication from the console supporting software **6**. In one embodiment, a successfully login on the control management station **2** is considered an authentication because it authorizes the user named "A" access to the CCDSVM. Thereafter, the user "A" can obtain all necessary information about the system units **3** and the control management station **2** from the console supporting software **6**. When the user A initiates a task for a selected target computing system, which is either a system unit **3** or the control management station **2**, the task information is transmitted via the net **11** or **12** from the web-console **9** to the console support software **6** on the control management station **2**.

A thread is created based on the console support software, **6** where the thread will serve and carry this task in the background. The created thread acquires a lock and stores the task information into a valid entry on a user level task list shown in FIG. **5**. This is one of efforts to ensure that each of the multiple tasks can be initiated simultaneously and concurrently within the same web-console **9** of FIG. **2** without delaying, effecting or blocking in the web-console **9** window and free from racing each other.

In addition, multiple web-consoles **9** for multiple concurrent users anywhere on the net **11** or **12** also can be supported. The obtained locks for this task will be properly released one at a time along with the task execution up to a point when the task is finally finished. Therefore, each task could be executed without time delay. Also, the stored task information will be removed upon the execution of the task is finished.

If total tasks initiated from the web-console **9** have succeeded the maximum tasks allowed by the console supporting software **6**, the initiated task is failed. The locks will be released by the corresponding thread and the user A on the web-console **9** will be notified correspondingly via net.

If an existing task is in a stage of changing a resource object on a target system and if a newly created task will make change on the same resource object on that target system, the newly initiated task may fail or may have to wait until the

previous task is finished. Further, if a task is failed, the locks associated with the task will be release by the thread and the user A on the web-console **9** will be notified across the network of **11** or **12** by the console support software **6**.

The credential of executing a specific task on a specific target computing system submitted by the user A is checked, where an ordinary users' access & operation permissions and credentials are setup by administrator with supervisor or special privileges. If the user A is not permitted to perform any task on such target computing system or is not permitted to perform such task on any one of the computing systems in the CCDSVM, the task execution will fail and the user A will be notified. Otherwise, the task will be carried out by the corresponding thread on the target computing system that is either a control management station **2** or a system unit **3**.

If there is a need, the console supporting software **6** will send the results or data back to the web-console **9**. When the task is failed nor succeeded, the threads of the console supporting software **6** will release the locks acquired for this task.

If the task needs to be executed on the control management station **2**, the thread created based on the console supporting software **6** will carry out this task. The threads of the console support software **6** also need to determine if they need to create another thread to execute this task. If there is a need, another thread will be created to execute this task. Once the task is finished, the corresponding locks will be released by the console supporting software **6**.

If a task needs to be executed on a system unit **3**, the console supporting software **6** will transmit the task information via the net2 **12** to the service software module **8** of the target system unit **3**. The thread based on the service software module **8** of the target system unit **3** will carry out this task. The service software module **8** on the target system unit **3** needs to determine if an additional thread needs to be created in order to execute such task. If there is a need, an additional thread is created to execute this task. Once the task is finished on the target system unit **3**, the corresponding status of the task execution is transmitted back to the console supporting software **6** of the control management station **2**. Upon receiving the task finished status, the locks associated with the thread of the console support software **6** for that task are released.

The Task Issued from Web-Console

The multiple concurrent tasks issued from a web-console **9** by a user could be any of the followings:

a) Move or transmit data such as a multiple gigabytes of file or other data in any form from any point or any computing system to another point or another computing system within the CCDSVM.

b) Configure, partition and assign entire storage system (raid/disk) within the CCDSVM.

c) Setup authentication for a specific user from a web-console on a specific console host with certain privilege for the entire CCDSVM or for a specific computing system, which could be any one of the system units **3** or a control management station **2**. Setting up the steps of authentication process for any specific services configured in one or more specific system units **3**.

d) Monitor and display activities and status for networks, storages, CPUs, processes and threads in the CCDSVM.

e) Create file system, file and directory structures, and support all other related data file operations on either the control management system **2** or the system units **3**.

f) And all other types of tasks and operations that might be run in other OS (operating system) environment.

The capability of providing user with the multiple concurrent simultaneous operations and tasks on the web console **9** has indicated that this invention has created a web-based user work environment of an operating system for a single computing system or for multiple computing systems. Further, this is a consistent working environment for the operating system for a computing system since it allows a user access to exact the same working environment through the web-console **9**, which could be a web browser residing either in the computing system or residing in a remote systems.

User Login

The user-login mechanism is also supported by the console supporting software **6**. The web-console **9** obtains a login web page from the console supporting software **6** via the network of **11** & **12**. Once the user provides an account name and a password via the login page displayed in the web-console **9** window, the authentication information is sent to the console supporting software **6** for validation. Upon successful validating the user account and password information, the console support software **6** sends all necessary information such as IP address to the web-console **9**, where the information also includes the information of the control management station **2** and system units **3**.

The Maximum Tasks

The maximum multiple concurrent simultaneous tasks that can be initiated from the web-consoles **9** are determined by the console support software modules **6**, and they are also determined based on the needs and the capacity of the control management station **2**.

The Credential Checking

The credential of a user includes the permission to access all or partial computing systems or a single computing system within the CCDSVM. The credential further includes the permission to run all tasks or partial tasks that are listed in the previous section of "The Task Issued From Web-Console". It also includes the permission of accessing a specific size of storage volumes. For example, a user B may be granted a permission to run tasks over computing systems X, Y, and Z. Another user C may be granted a permission to run tasks over the entire computing systems in the CCDSVM environment. The user C might be allowed to get system status on the computing systems X, Y, and Z only while the user B may be allowed to run all tasks on the computing systems X, Y, and Z. Each computing system mentioned here could be a control management station **2** or any of system units **3**. This basically represents a two-level authentication policy and checking. The first level is the security imposed on the control management station **2** and the second level is the security imposed on the system units (**3** of FIG. **2**).

The Web-Server Interface Software Modules

The web-server interfacing software module **5** is responsible to get information from or send information to the web server software **7**. It also interacts with the control management modules **4** via inter-process communication and communicates with service module **8** of the system unit **3** via the net2 **12**.

The Control Management Software Modules

The control management modules **4** on the control management station **2** are responsible for communicating with the system units **3** for sending data to or receiving data from the system units via the net2 **12**. It also provides information of the system units **3** to the web interface software modules **5** of the control management station **2** via an inter-process communication mechanism.

The Layered CCDSVM Structure

To be more efficiently supporting multiple concurrent tasks over a larger number of the system units **3**, the CCDSVM can be organized into a multi-layered structure as illustrated in FIG. **6**. With this layered structure, the CCDSVM can be sub-divided into different groups. For example, each one of

level-2 control management stations could function both as the control management station **2** for controlling the system units **3** below it and as a system unit **3** that is controlled by the level-1 control management station **2**. Therefore, the level-2 control management station must be configure with related software modules for both the control management station **2** and the system unit **3**.

What is claimed is:

1. A server supporting access to resources, the server comprising: at least one hardware processor, and a network interface; wherein the server is configure to perform followings:

causing display of information about resources organized by the server in a web browser on a first end-user device;

receiving a first request, for access a first resource, from the first end-user device upon a first user selecting the first resource from the information displayed on the first end-user device and submitting the first request;

storing information about the first request and invoking a lock protection to protect the storing of the first request;

processing the first request, including to process the first request in the background and cause the display of the information about the resources without blocking in the web browser during a regular network traffic to allow the first user selecting a second resource from the information displayed on the first end-user device and submitting a second request for access to the second resource without waiting for the completion of the first request; and deleting the stored information about the first request when the first request is completed.

2. The server of claim **1**, further comprising:

allowing a second user from a second end-user device access to at least one of the resources in the same way as to allow the first user from the first end-user device access to the first and second resources.

3. The server of claim **2**, wherein the information about the resources organized by the server reside in the server or are collected by the server from one or more computing devices across a network.

4. The server of claim **1**, further comprising:

receiving the second request, storing information about the second request and processing the second request in the same way as to process the first request, and deleting the stored information about the second request when the second request is completed.

5. The server of claim **1**, wherein said processing of the first request further comprises:

obtaining a response comprising status or result of the processing of the first request and causing display of the response in the web browser on the first end-user device.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 8,799,473 B2
APPLICATION NO. : 12/075314
DATED : August 5, 2014
INVENTOR(S) : Sheng Tai Ted Tsao

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In the Specification

In col. 1, line 8 - 9, please delete "patent application No. 12/075,314 filed on 03/04/2008, which itself in turn is a continuation application of US"

Signed and Sealed this
Second Day of December, 2014

Michelle K. Lee

Michelle K. Lee
Deputy Director of the United States Patent and Trademark Office

Exhibit E

US009280547B2

(12) **United States Patent**
Tsao

(10) **Patent No.: US 9,280,547 B2**
(45) **Date of Patent: *Mar. 8, 2016**

(54) **SYSTEM AND METHOD FOR DISPLAYING AND OPERATING MULTI-LAYERED ITEM LIST IN BROWSER WITH SUPPORTING OF CONCURRENT USERS**

(71) Applicant: **Sheng Tai Tsao**, Fremont, CA (US)

(72) Inventor: **Sheng Tai Tsao**, Fremont, CA (US)

(73) Assignee: **Sheng Tai (Ted) Tsao**, Fremont, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 374 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **13/914,515**

(22) Filed: **Jun. 10, 2013**

(65) **Prior Publication Data**

US 2014/0365975 A1 Dec. 11, 2014

(51) **Int. Cl.**

G06F 17/30	(2006.01)
H04L 12/24	(2006.01)
H04L 29/06	(2006.01)
H04L 29/08	(2006.01)
H04L 12/26	(2006.01)

(52) **U.S. Cl.**

CPC ***G06F 17/30126*** (2013.01); ***H04L 12/2458*** (2013.01); ***H04L 41/0253*** (2013.01); ***H04L 41/22*** (2013.01); ***H04L 65/403*** (2013.01); ***H04L 67/1097*** (2013.01); *H04L 12/24* (2013.01); *H04L 12/2424* (2013.01); *H04L 12/2602* (2013.01); *H04L 29/08099* (2013.01); *H04L 41/0213* (2013.01); *H04L 41/0266* (2013.01); *H04L 41/082* (2013.01); *H04L 41/0806* (2013.01); *H04L 41/0893* (2013.01); *H04L 41/12* (2013.01); *H04L 67/02* (2013.01)

(58) **Field of Classification Search**

CPC H04L 29/08099; H04L 41/22; H04L 12/2602; H04L 12/2408; H04L 12/2458; H04L 12/24; H04L 12/2424; H04L 67/1097; H04L 67/02; H04L 65/403

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,047,495	B1 *	5/2006	Pang	G06F 8/38 715/736
7,058,696	B1 *	6/2006	Phillips	G06F 17/30861 707/999.01
7,222,147	B1 *	5/2007	Black	H04L 41/0803 709/200
7,363,359	B1 *	4/2008	Tripathy	H04L 41/20 370/350
8,073,954	B1 *	12/2011	Tu	H04L 29/06 709/203
2002/0091697	A1 *	7/2002	Huang	G06F 17/30867
2002/0112076	A1 *	8/2002	Rueda	H04L 12/14 709/245
2002/0129096	A1 *	9/2002	Mansour	G06F 9/4443 709/203

(Continued)

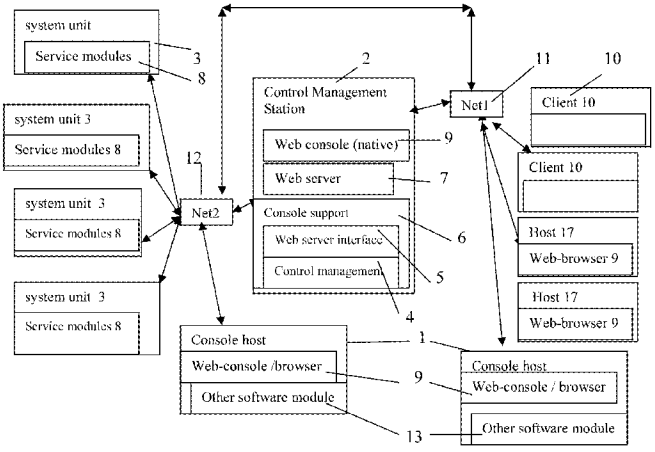
Primary Examiner — Tom Y Chang

(57) **ABSTRACT**

Support of end-user to view and operate computing resources through logically organized and graphically represented multi-layered item list ("hierarchical list") has been fully realized on native window based computer user work environment with modern operating system such as Windows Explore of Microsoft. The present invention has implemented such a hierarchical list to represent structured resources, such as for a central controlled distributed scalable virtual machine (CCDSVM), and to be displayable and operable via a browser on an end-user device for user to access and manage the actual structured resources. The implementation of the hierarchical list is accomplished by creating the hierarchical list in memory to mirror the actual structure resource, where the mirrored hierarchical list is sent to the end-user device, where the end-user device executes the browser to display the hierarchical list to allow the user to access the actual structured resources via the displayed hierarchical list.

20 Claims, 13 Drawing Sheets

A Typical CCDSVM with Multiple Clients



(56) **References Cited**

U.S. PATENT DOCUMENTS

2002/0133697 A1* 9/2002 Royer G06F 21/41 713/150
2003/0079032 A1* 4/2003 Orsolits H04L 12/2602 709/230
2003/0167229 A1* 9/2003 Ludwig G06Q 30/04 705/40
2003/0229900 A1* 12/2003 Reisman G06F 17/30873 725/87
2004/0003091 A1* 1/2004 Coulthard G06F 9/544 709/227
2004/0006614 A1* 1/2004 DiFalco H04L 41/0893 709/223
2004/0133650 A1* 7/2004 Miloushev G06F 17/30197 709/213
2004/0133652 A1* 7/2004 Miloushev G06F 11/1076 709/214
2004/0181578 A1* 9/2004 Elms H04L 67/14 709/205
2004/0205473 A1* 10/2004 Fisher G06F 21/41 715/255
2004/0255289 A1* 12/2004 Alex George G06F 9/4445 717/174
2005/0216860 A1* 9/2005 Petrov H04L 12/2602 715/810
2005/0229154 A1* 10/2005 Hiew G06F 8/34 717/110
2006/0202964 A1* 9/2006 Liaw H04L 12/24 345/168

* cited by examiner

Fig. 1: A Typical CCDSVM with Multiple Clients

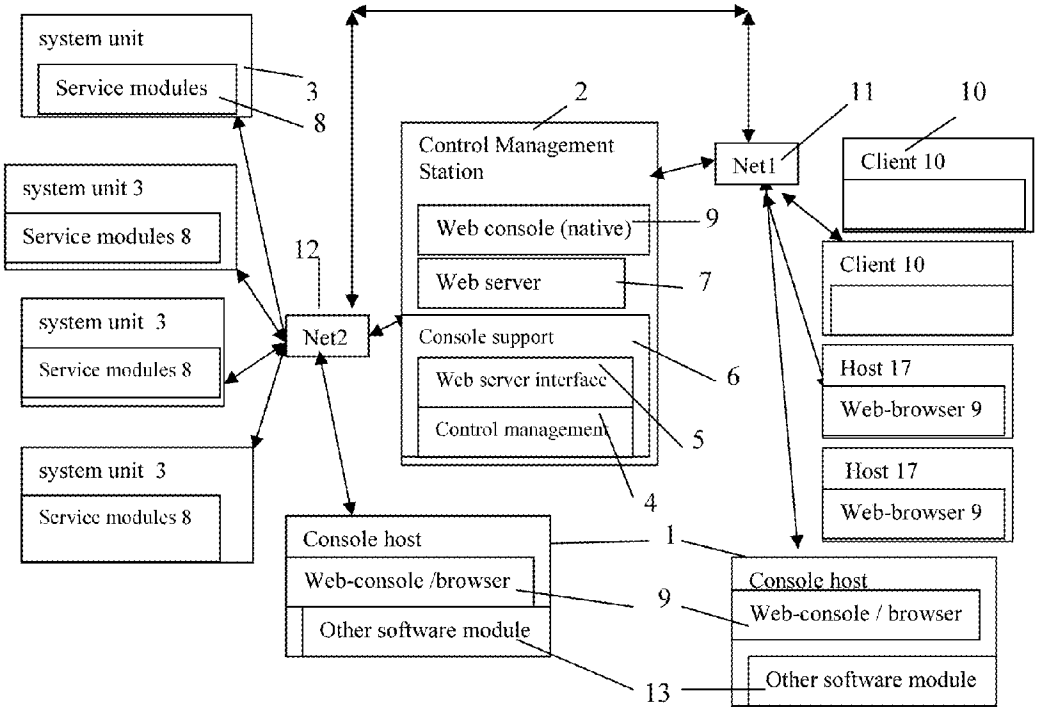


Fig. 1

Fig. 2 a) An example of a degenerated CCDSVM with a control system only

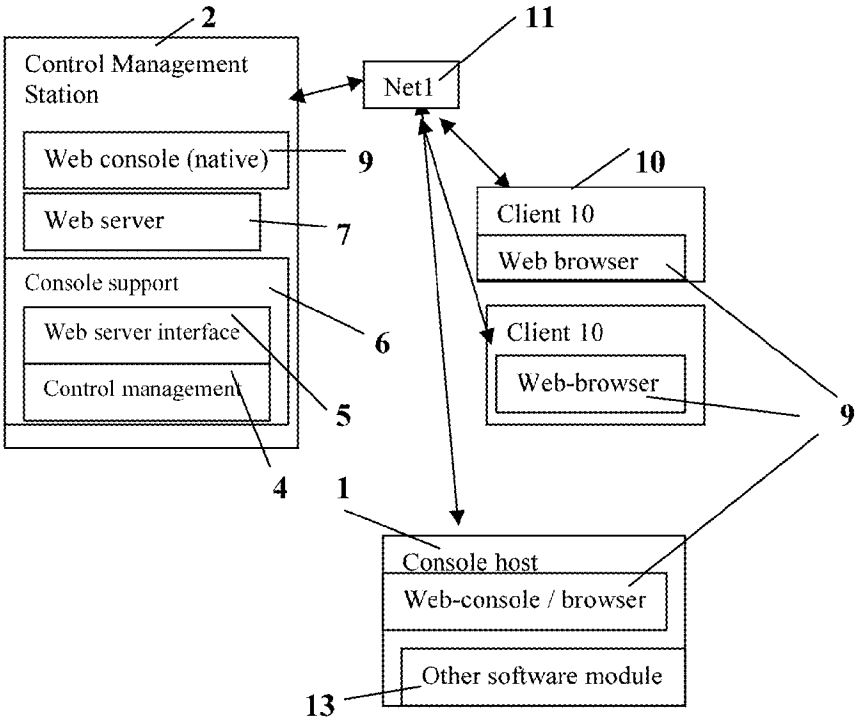


Fig. 2 a)

Fig. 2 b) An example of a CCDSVM with identical system unit and client host

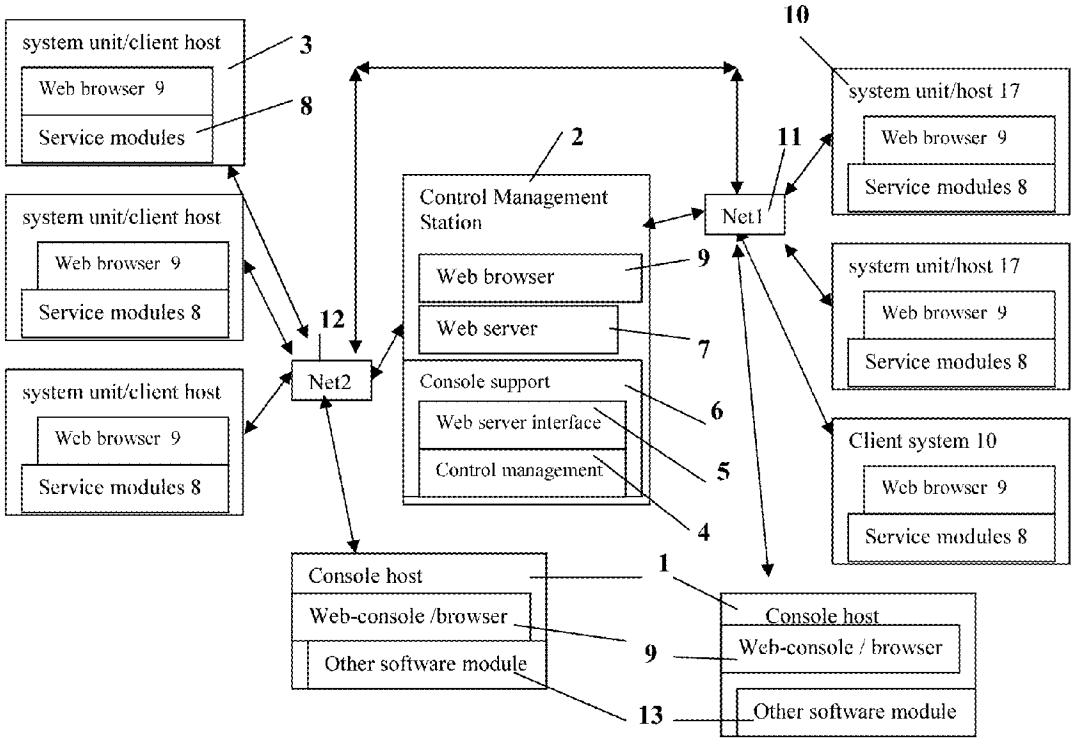


Fig. 2 b)

Fig. 2 c) An example of a CCDSVM for non-web based client access and web-based control management

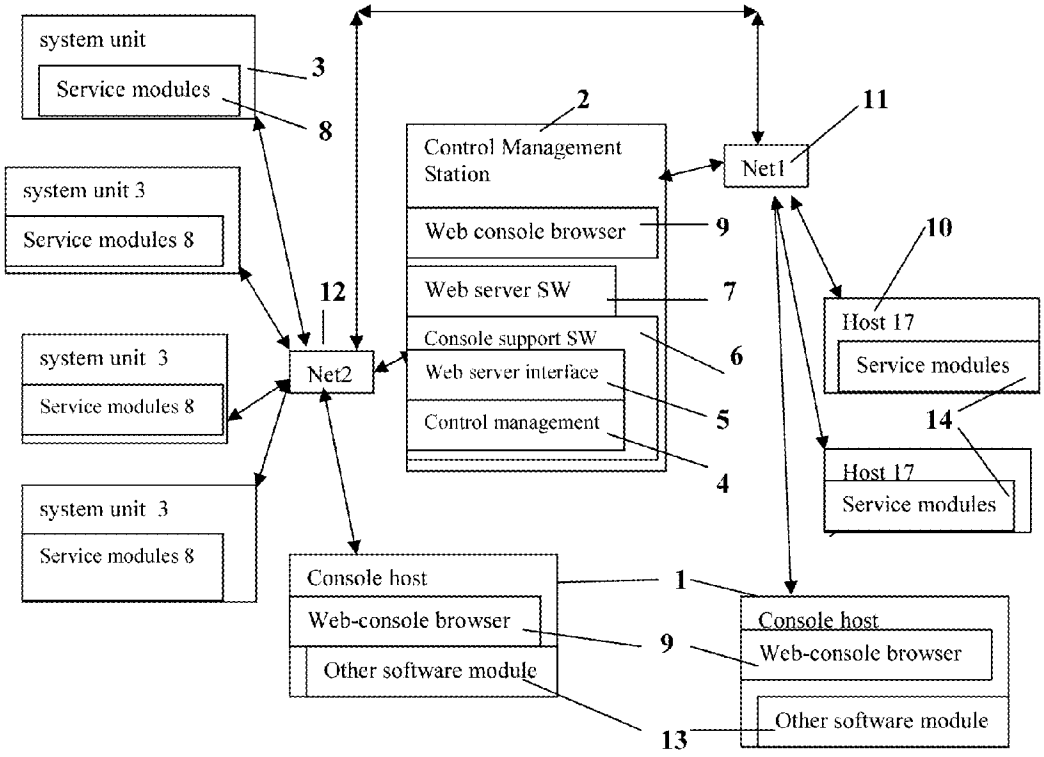


Fig. 2 c)

Fig. 3) The Software of WCUWE for a typical CCDSVM:

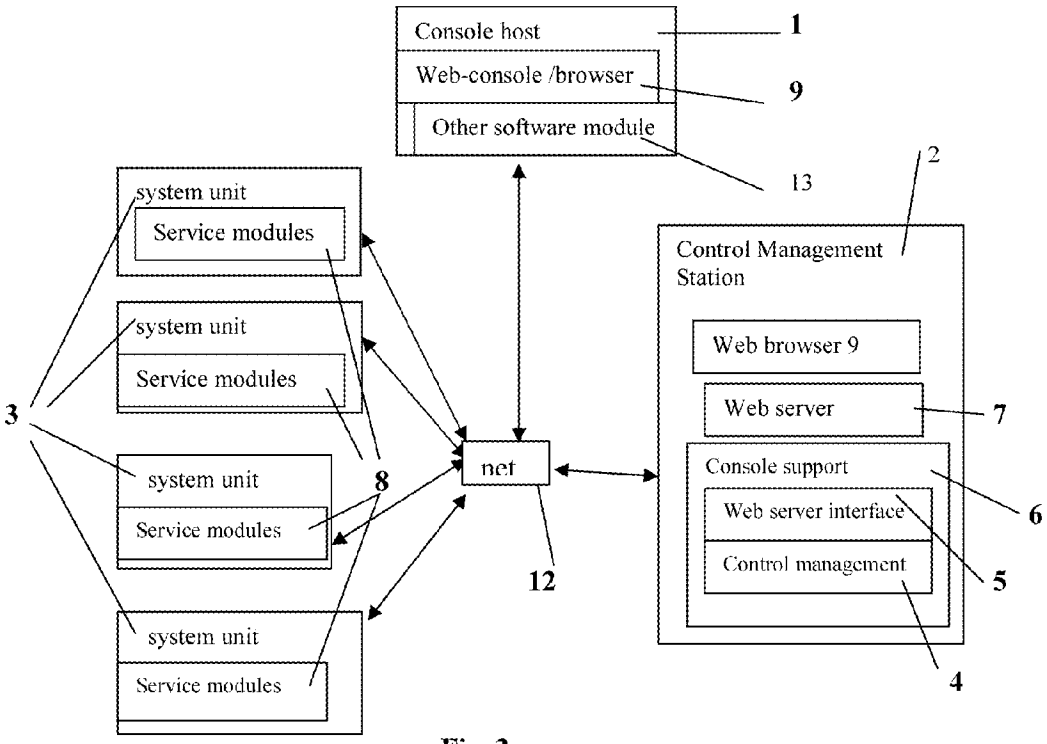
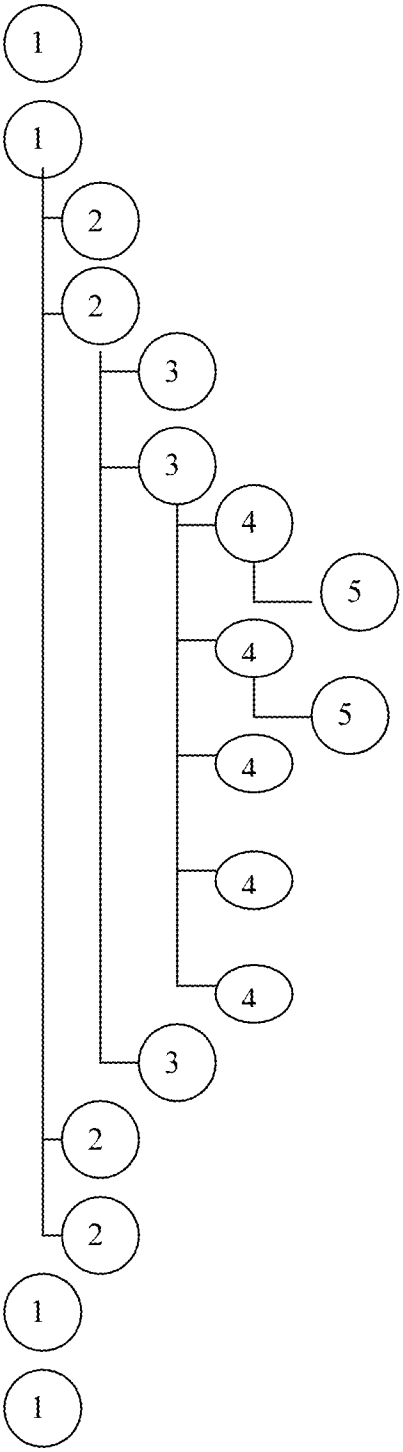


Fig. 3

Fig. 4: Examples of Multi-Layer Items List:

4a)

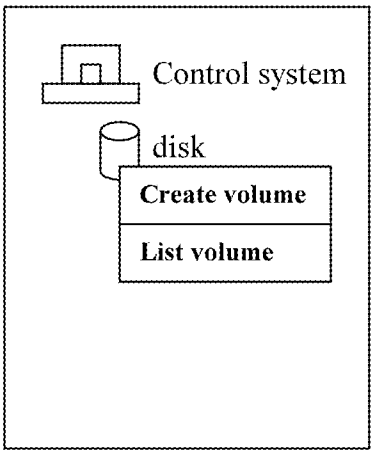


4b)

Fig. 4

Fig. 5, Examples of web based operation menu

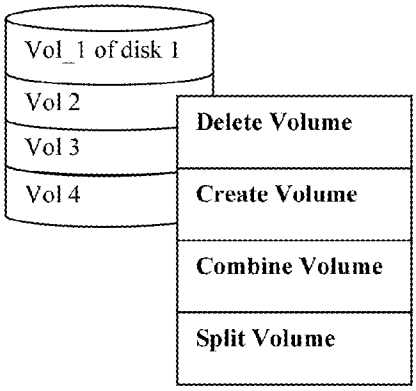
5 a) Web based drop down menu for disk operation.



5 b) Web based selective menu for selecting an system unit:

Fig. 6: Examples of web based operation menu for different type of resource node on a MLIL.

6 a) an operation menu for managing a disk storage.



6 b) an operation menu for managing files and folders.

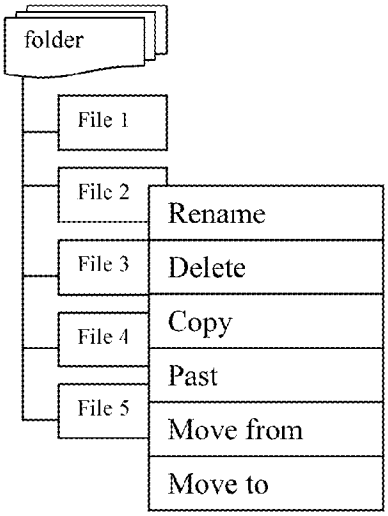


Fig. 6 C) a web based operation menu for managing system units on network.

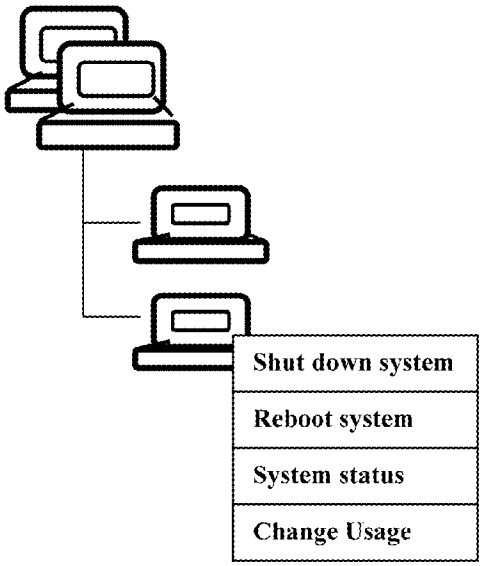


Fig. 6 C

Fig. 7: An example of system units are automatically and dynamically grouped by using group ID.

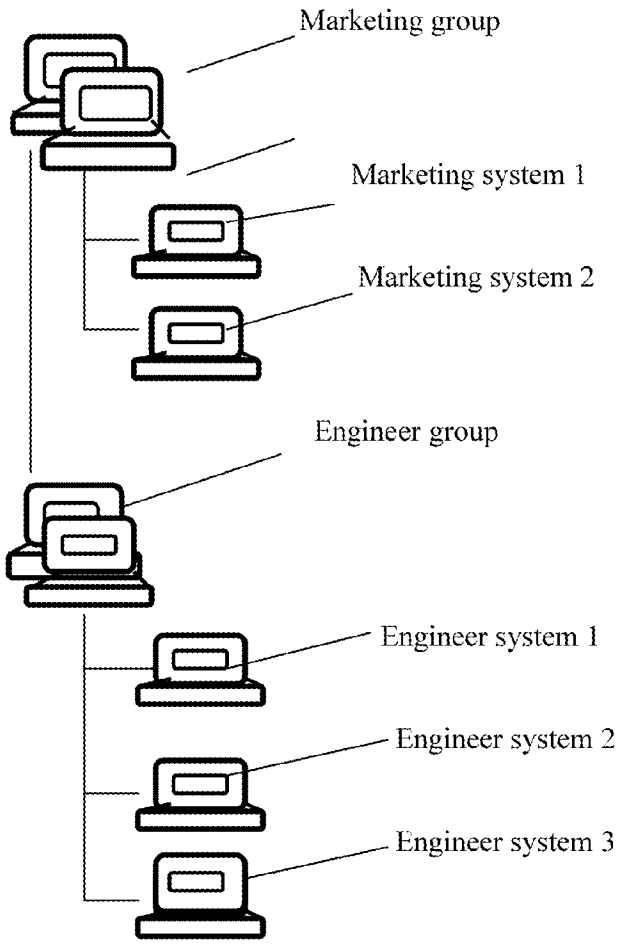


Fig. 8 a): An example of a MLIL representing a structure combined with mixed devices, system units and groups.

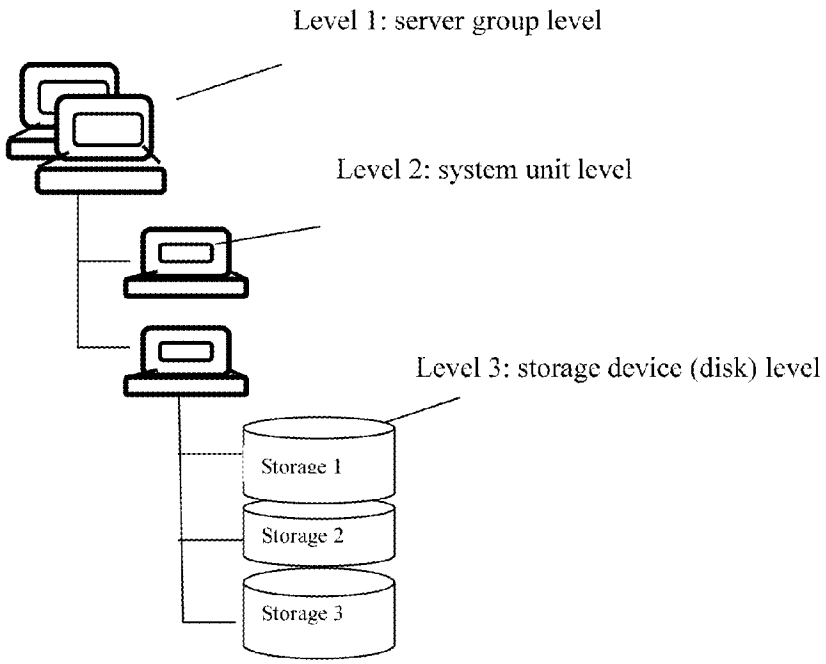


Fig. 8 b) This example illustrates that the top level of a MLIL is for system group, the second level of the MLIL is for system unit, and folders and files start from the third level of the MLIL.

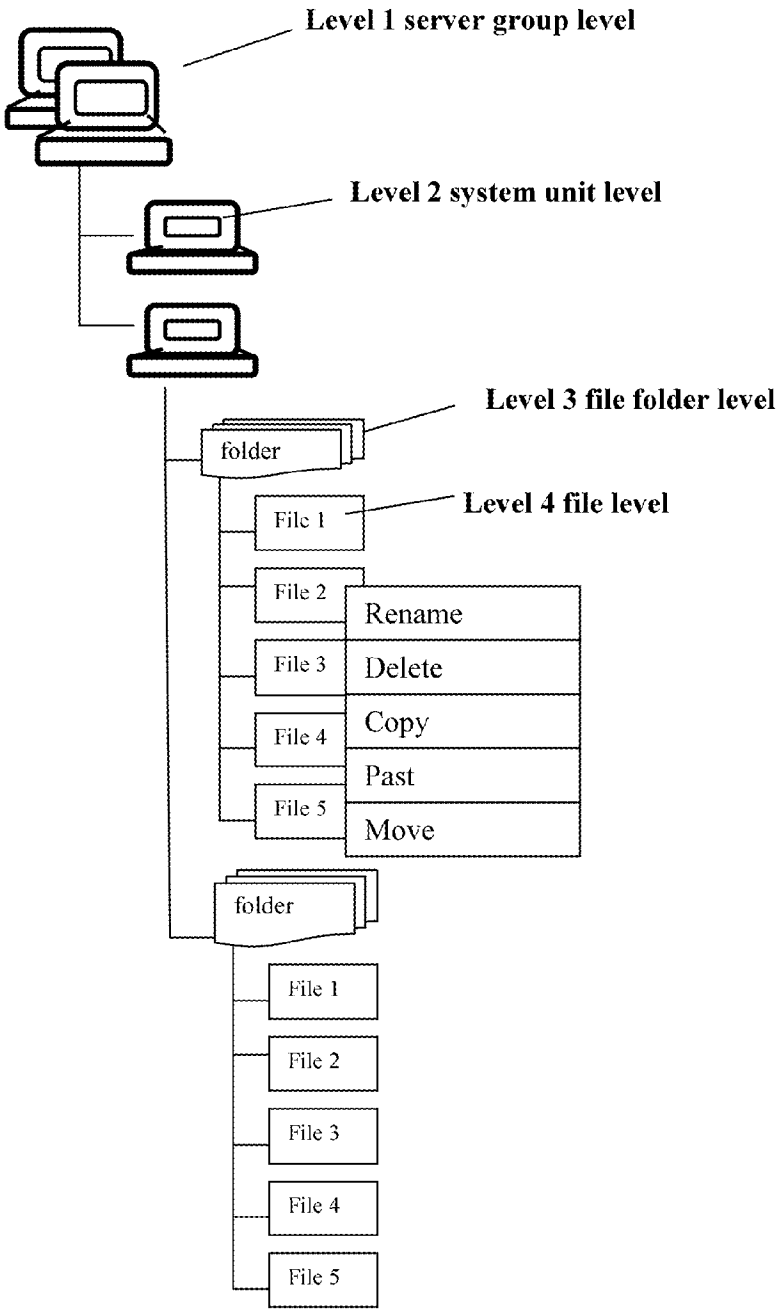
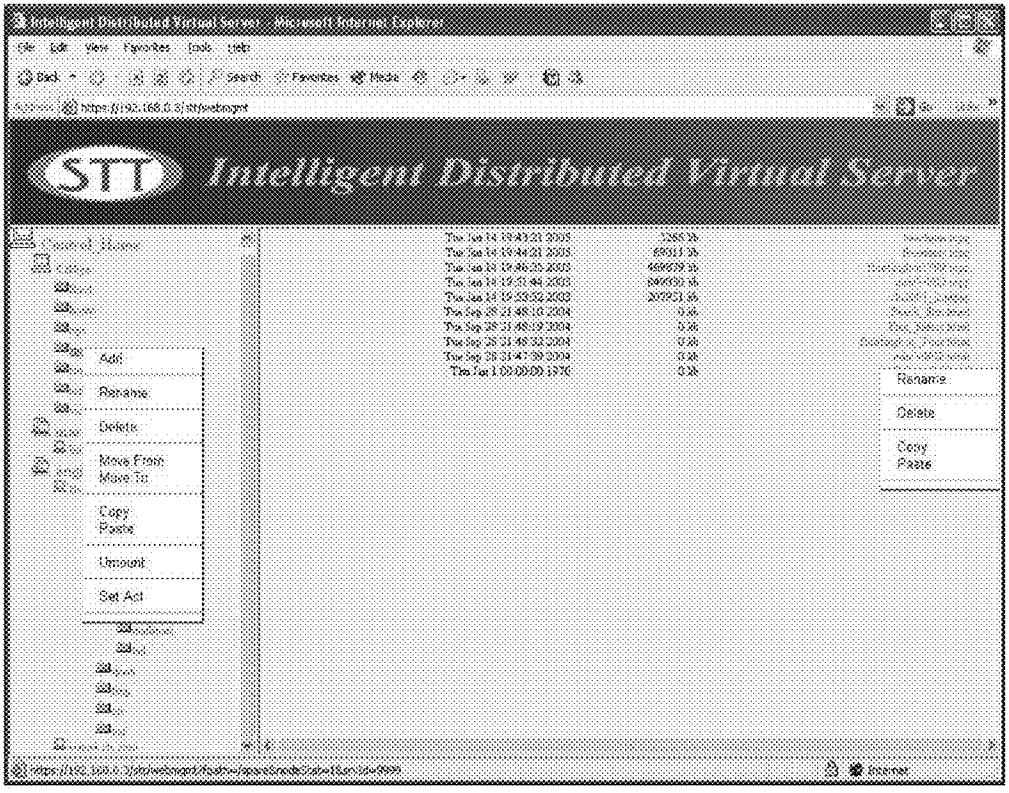


Fig. 9 This figure illustrates one embodiment of how does the file system on a system unit in a system group of the virtual server can be accessed.



SYSTEM AND METHOD FOR DISPLAYING AND OPERATING MULTI-LAYERED ITEM LIST IN BROWSER WITH SUPPORTING OF CONCURRENT USERS

PRIORITY AND CROSS REFERENCES

This is a continuation application of U.S. patent application Ser. No. 13/050,871 filed on Mar. 17, 2011, which in turn is a continuation application of Ser. No. 12/709,434 filed on Feb. 19, 2010, which in turn is a divisional application of U.S. patent application Ser. No. 11/374,302 filed on Jul. 2, 2004 that is converted from provisional application 60/585,552, and now is a U.S. Pat. No. 7,945,652. The application Ser. No. 11/374,302 is a continuation-in-part of U.S. patent application Ser. No. 10/713,904, filed on Aug. 6, 2002 that is converted from U.S. provisional application No. 60/401,238, and now is a U.S. Pat. No. 7,418,702. The application Ser. No. 11/374,302 is also a continuation-in-part of U.S. patent application Ser. No. 10/713,905, filed on Aug. 12, 2002, that is converted from U.S. provisional application Ser. No. 60/402,626, and now is a U.S. Pat. No. 7,379,990. All above mentioned applications are herein incorporated by reference in their entireties for all purposes.

BACKGROUND OF THE INVENTION

1. Field of Invention

The present invention is in the area of network communication, and particularly related to multiple users concurrently interacting with multiple computers to display, view, access, operate various resources (e.g., via multi-layers item list) by using a web browser In addition, the present invention is also related to methods and steps of constructing the web-based computer user work environment.

2. Description of Related Art

The evolution of computer-related working environment has changed from paper tape or punch cards of early years to a command line user work environment on a native system in 70th, and to a graphic window environment on a native system in the middle 80th. Since then, there have been some network-based computing environments developed such as Microsoft's terminal service, the Telnet of Unix & Linux system, and etc. These computing environments allow users to access and operate a target system from another remote system. Some network management software has also been developed to mostly focus on monitoring the activities of a target system. However, most of the software are limited in mobility and capability because they need to install the specialized software on the remote system.

There are some web-based application software that allow users to access web applications on a web server from a web browser running on a remote system or device. However, most of such application software are not designed for managing resources on a web server and/or for creating a web-based computer user working environment. There are also some web-based software that allow people to manage a system remotely. Again such software is not designed to create a web-based computer user working environment instead of dumping the existing computer user work environment of a system to a web browser on a remote system. Thus there is a need for the user-friendly environment that users can concurrently access and manage resources with ease.

SUMMARY

This section summarizes some aspects of the present disclosure and briefly introduces some aspects in preferred embodiments. Simplifications or omissions in this section as well as in the abstract or the title of the present disclosure may be made to avoid obscuring the purpose of this section, the abstract and the title. Such simplifications or omissions are not intended to limit the scope of the present disclosure.

Generally speaking, the present invention is related to a user-friendly environment that users can concurrently access and manage resources with ease. According to one aspect of the present invention, a web-based computer user working environment (herein "WCUWE") is disclosed. Some aspects of the WCUWE are described in co-pending U.S. application Ser. Nos. 10/713,904 and 10/713, 905, both of which are hereby incorporated by reference in each one's entirety. A central-controlled distributed scalable virtual machine (herein "CCDSVM") is formed to achieve and realize a flexible working environment for users of the CCDSVM. The WCUWE provides a mechanism to allow users to access and operate the provisioned systems of a CCDSVM from any web browser on any remote system or devices. With the capability of expanding a CCDSVM across the Intranet and Internet with its strong security control, the WCUWE can meet the increasing demand for a more secure, more mobilized, more flexible and more cost saving computational solution.

BRIEF DESCRIPTION OF THE FIGURES

These and other features, aspects, and advantages of the present invention will become better understood with regard to the following description, appended claims, and accompanying drawings where:

FIG. **1** shows a system view of a CCDSVM according to one embodiment of the present invention;

FIG. **2***a* shows an exemplary CCDSVM with a single control management station;

FIG. **2***b* shows an exemplary CCDSVM with identical client host and system unit;

FIG. **2***c* shows an exemplary CCDSVM with client hosts for non-web based accessing;

FIG. **3** is an example of functional block diagram of a WCUWE for the CCDSVM;

FIG. **4** shows a simplified example of a multi-layered item list;

FIG. **5** shows some examples of an operation menu for WCUWE;

FIG. **6** shows an example of an operation menu for different types of nodes;

FIG. **7** shows an example of server units grouped by a group ID;

FIG. **8** shows examples of a multi-layered item list with items from other devices, system units or groups; and

FIG. **9** illustrates an example of how does the file systems on a system unit in a system group can be accessed.

DETAILED DESCRIPTION OF THE INVENTION

The detailed description of the invention is presented largely in terms of procedures, steps, logic blocks, processing, and other symbolic representations that directly or indirectly resemble the operations of data processing devices. These process descriptions and representations are typically used by those skilled in the art to most effectively convey the substance of their work to others skilled in the art. Numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will become obvious to those skilled in the art that the invention may be practiced without these specific details. In other instances, well known methods, procedures, components,

and circuitry have not been described in detail to avoid unnecessarily obscuring aspects of the present invention.

Reference herein to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment can be included in at least one embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment, nor are separate or alternative embodiments mutually exclusive of other embodiments.

For the convenience of describing some embodiment of the present invention, the following terminologies are provided within the scope of present invention.

CCDSVM

is an abbreviation for central controlled distributed scalable virtual machine. A CCDSVM is configured to allow a control management station to control a group of systems (e.g., server computers) and provide distributed services to a client device coupled to the Intranet, the Internet or in a local area network (LAN) environment. The software components of the CCDSVM form a virtual operating environment or system.

Every computer system provides an end user a working environment (CUWE), which usually runs on top of a generic operating system. The CUWE allows a user to setup an authentication profile for each different user, to configure system resources from storage, network, file system, file folders structure, files and all other available resources on the system, to monitor system activities, to access applications, and to manage data files such as moving data from one folder to another folder or moving data from one system to another system and etc without limits. Specially, a CUWE is configured to provide a multi-task mechanism. Therefore, a user can perform multiple tasks concurrently after they successful log in the CUWE.

The web-based computer user work environment (WCUWE) of present invention runs on top of a generic operating system in a single or multiple computers. It allows one or more users to work on one or more computers through a conventional web-browser either remotely or locally, from a device such as a laptop computer, a PDA, or a desktop computer. It provides great mobility and flexibility to the user(s). With this web-based computer user work environment, the entire CCDSVM system can operate like a single virtual machine.

In general, resources on a computer or any other electronic device refer to, but not limited to, CPU, memory, storage device, network device, monitor or other display devices, keyboard, mouse, photo or video record devices, wireless device, user authentication profiles, and various form of deposited data in a storage device. The deposited data in a storage device could be a file system structure, folders, data files, formed raw data and etc. and is not limited to these in the present invention. A data file can be in all kinds of form, such as plain text form, binary data form, PDF form, MPEG or JPEG form, or other various image data forms, MS power point data form or Word documentation form etc. without limits. The storage device could be in any media form such as hard disk drive, magnetic tape drive, various form of memory devices, or others suitable media without limits. it should be noted, unless specifically discussed, whenever a resource in a computer, a system or a device is discussed, it means any form of computing power, data, or capacity the computer, the system or the device has or able to access.

A web browser can interpret data in a standard structured format (formatted information) and display the data as web pages so that people can view, manipulate, and interact with the information therein. The standard formatted information

is imposed and encoded by following the syntax of a programming language such as HTML (Hypertext Markup Language), XHTML, DHTML, XML, or any other various suitable languages, whichever is best to describe the structured information without limits. In practice, these languages (HTML, XML, . . .) themselves are often referred to as a standard for web page programming. Therefor, in this disclosure, it is not necessary to specify what standard is being used as long as data or resources are presented in a way that is suitable for web communication.

A multi-layered item list (MLIL) is a logically organized information list with certain order. Each entry in an MLIL contains an item that may contain another layered list of items, where each of the items may represent a physical resource or information about a computer system or other devices. For example, the information of files and folders of a file system on a modern computer system is typically organized as a multi-layers item list and can be entirely viewed, displayed and operated in a UI display on a native system with supporting from a modern computer operating system. Likewise, each item (also referred to as a node) on an MLIL may represent a folder or a file. According to one embodiment of the present invention, the information about systems or devices on a network, the information of hardware components on a system such as disks, network cards, memory etc, the information of users of a system can be organized into an MLIL for view, display and various other operations. The supporting of the MLIL in a WCUWE makes the system resources and information much easy to be displayed, viewed, and operated in a browser. In general, an MLIL contains at least one item (node) and as many layers as needed.

The system mentioned in this invention is any type of computing device that can be a desktop computer, laptop computer, various types of servers, PDA, or cell phone or other devices with communication ability across a communication network.

The operating system (OS) mentioned in this invention can be any suitable operating system such as Windows, Linux, various Unix, real-time operating system and others without limits.

The programming languages, which used for implementing all software mentioned in this invention, could be any suitable languages or a combination of the suitable languages such as C, C++, Java, JavaScript, Visual Basic, C sharp, HTML, XML, DHTML, XHTML, and others without limits

The communication protocols to be used in the CCDSVM could be various types that are appropriate for transmitting required data across a communication network. The communication protocols could be IP-based or non-IP-based. The IP based protocols are built on top of IP protocol that could be standard protocols such as TCP protocol, UDP protocol, ICMP protocol, and others without limits. The IP based protocols also can be non-standard proprietary protocols. The non-IP based protocols can be ISO 8473, ISO 8208, or serial communication, or data link layer protocols like LLC 802.2, or HDLC, or any proprietary protocols bellow the IP protocol level.

The communication protocols for web computing in the present invention could be HTTP, SOAP, WAP, or others without limits.

The web browser could be any existing commercial software from any vendor such as Microsoft IE, Netscape, Firefox, Mozillar, or other commercial or proprietary software. The web browser must be capable to handle web protocols

such as HTTP, SAOP, WAP or others, and be able to interpret the standard structured formatted information such as web page.

The web server software mentioned in this invention could be a commercial software from any vendor such as Apache of open source foundation, IIS of Microsoft Corporation, or others on the market, and it also can be a proprietary software. The web server software must be able to handle web protocols such as HTTP, which is a protocol built on top of TCP protocol. Therefore, the structured information for web communication can be transmitted to mentioned web browser across a communication network by the web server if the web server and web browser are located on two different systems. The web server and web browser also can communicate with each other through inter-process communication if both of the web server and web browser are located on a same computing system.

A user session with the CCDSVM is started at a time a user login (log on) to a CCDSVM from a web-browser via an end user computing device and is ended at a time the user logout the CCDSVM from the same web browser. During a session, users can perform tasks which have been permitted and specified in a security profile for the user.

One of the objects, advantages, and benefits in the present invention is to provide a web based computer user working environment ("WCUWE") in which users can access resources and data available in an environment including a plurality of resources and data resided on a plurality of computing systems. According to one embodiment of the present invention, such resources and data can be accessed and managed from a web-browser regardless where the users are located. In addition, such an environment permits each of multiple users from his or her own web-browser resided on a computing system or device to concurrently access and manage the CCDSVM with multi-tasking capability.

Since the introduction of graphic user interface in an operating system for computers, users are able to display, view and operate data and hardware resources in a computer system or device by a pointing device (e.g., a mouse). By clicking through a multi-layered item list, such as a file or folder list displayed in Window Explore in Microsoft Windows OS, a user can navigate to an item in any level in the list. In other words, supporting a multi-layered item list has become a critical part of a native window based computer user work environment. However, within the CCDSVM, how to let multiple concurrent users to effectively simultaneously display, view or operate an item list in a single web-browser for the available resources or data, such as for deeply nested files & folders or for multi groups of provisioned systems or devices etc. is an unprecedented challenge.

Referring now to FIG. **1**, it shows an exemplary simplified block diagram of an exemplary CCDSVM according to one embodiment of the present invention. As shown in the figure, the CCDSVM is a collection of computing devices including console hosts **1**, a control management station **2**, system units **3**, and client hosts **10**, all communicating via net**1** **10** and net**2** **11**. Each of the elements in the CCDSVM is described in detail below with respect to FIG. **1**.

A console host **1** can be any computing system on a network and run a suitable OS. However, it must have a web browser **9** installed, where the browser **9** is used for accessing and operating the entire CCDSVM. The web-browser **9** can be implemented with any suitable or a combination of suitable programming languages. One example of the browser **9** is the Internet Explorer (IE) from Microsoft Corporation. Optionally, Web-Console is another name for the web-browser **9** on a console host **1**. From the web browser **9** by following a web URL link to another computing system or on the same system, a user can obtain a WCUWE and further access, manage and operate the entire system. The term of the web-console used sometimes is to indicate its ability to access system information and perform system operation in the CCDSVM by privileged or authorized users. A console host **1** may also include other software modules **13**, which may be implemented with any suitable programming languages. These software modules **13** may communicate with a control management station **2** using IP based, non-IP based protocols or any suitable protocols to receive or send data between a console host and the control management station **2**. To support a non-web-based networked console, the software used for the console must handle protocols other than web protocols such as HTTP, SOAP, WAP without limits. Further software modules **13** of the console host **1** must communicate with console supporting software **6** on the control management station **2**. The console host **1** is similar to the client system **10** except it is dedicated to a user who takes a system administration role for the CCDSVM.

The control management station **2** is an any type of computing system on the network and can have any suitable OS running on it. The control management station **2** includes web server software **7** and console supporting software **6**. The console supporting software **6** includes web server interface software modules **5** and control management software modules **4**. The control management station **2** may also have a Web browser used as a web-console **9**. The web server software **7** is configured to send data to and receive data from the web console **9** of one of the console hosts **1** or from a web-browser of client **10**.

The console supporting software **6** and the web server **7** can be implemented with any suitable or a combination of suitable programming languages. In one embodiment, the web server interface **5** of the console support software **6** may be extended to provide service as the web server software **7** does. In this case, there is no specialized web server software **7** needed.

The system unit **3** could be any type of computing system having a suitable OS running and containing service software modules **8** that is configured to communicate with other computing system (device) on the network. For example, the service software modules **8** of the system unit **3** can communicate with the control management software **4** of the control management station **2** to carry out tasks for viewing or operating its resource. It also can communicate with the client system **10** to deliver the required services, or to communicate with another system unit **3** to transfer data and etc.

The service software modules **8** may include an individual software module configured to handle the HTTP protocol or other web protocols if there is a need for web-based communication with the client system **10** or other system unit **3** or the control management station **2**. This individual software module could be commercially available web server software on the market or could be a proprietary software module.

Net**1** **11** and Net**2** **12**: each of Net**1** and Net **2** is a network infrastructure including the Internet or the Intranet, or a local area network (LAN) and can be a wireless, wired network. or a combination of wireless and wired network. They provide the necessary means to communicate among the devices in the CCDSVM. In one embodiment, Net **1** provides communication links between the control management station **2** and the console-host **1** or the client system **10**. It also provides communication link between the system unit **3** and the client system **10**. The net**1** may comprise connection media such as connecting cable of Ethernet, optical Fibre, and other; wireless communication media for providing wireless communi-

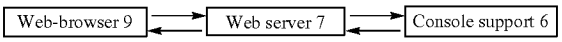
cation links through air; and data bus on circuit board. The Net**1** also consists of communication equipment such as switches, routers, and adapters, etc. and all other possible elements of communication equipment without limits.

The net**2** (**12**) is also a network infrastructure of Internet or Intranet, or LAN, which comprises similar connection media and equipment mentioned above and provides communication links between the control management station **2** and the system units **3** or the consoles host **1**. It also provides the communication link between the system unit **3** and the client host **10**.

The client hosts **10** are not part of the CCDSVM, but with the permission and authorization, they can get services from and access to the CCDSVM using the web-browser **9**.

According to one embodiment, there are 4 basic data flows paths within a typical CCDSVM:

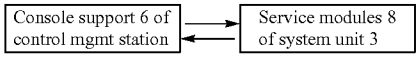
(1) Data path **1** is a data flow through a communication link between the web browser **9** on the client host **10** or the console host **1** or the control management station **2** and the web handling software on the control management station **2**, such as web-server **7** and the console support software **6**. With this path of data flow, whenever the user sends a request from the web-browser **9** to the web-server **7** and further down to the console support software **6**, the console support software **6** collects all required information (i.e., metadata) from a target system and converts them into standard structured information for web communication. The targeted system could be the system unit **3** or the control management station **2** itself. The information collected by the console support software **6** in the control management station **2** could be the system status, or the storage information, or the network information, or the user authentication profile, or the file system information or files and folders information on a target system. The collected information is not limited to these mentioned above herein. The console support software **6** then passes this converted structured information to the web server software **7** and further transmits them to the web-browser **9** through communication link, net**1** **11** or net**2** **12**, so that it can be displayed and viewed by the conventional web browser **9**.



The communication protocol used between the web-browser **9** in the client host **10** or the console host **1** and the web server **7** in the control management station **2** could be based on the HTTP or any other suitable protocols for web communication, which could successfully transmit the data over the web.

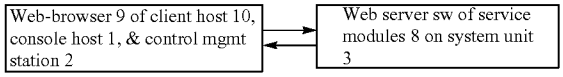
(2) The data path **2** is the data flow through communication link between the control management station **2** and the system units **3**.

With this path of data flow, the requests targeted to the system units **3** are passed from the console support software **6** in the control management station **2** to the service modules **8** in the system unit **3** through the communication link net**2** **12**. If the responses to those requests must return back to the control management station **2**, the service modules **8** in the system unit **3** carry out the requests and then send the response back to the console support software **6** in the control management station **2**. The communication protocol used between the console support software **6** in the control management station **2** and the service modules **8** of the system unit **3** can be any suitable protocol for transmitting data between them. The typical data flow through this path could be a boot message, a system status, network information, or storage information in the system unit **3**. It should be noted that the actual data flow though this path is not limited to those mentioned here.



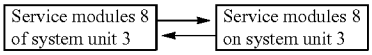
(3) The data path **3** is the data flow through a communication link between the system units **3** and the client host **10** or the console host **1**.

With reference to FIG. **1**, in some cases, the console support software **6** of the control management station **2** may present a web link, pointing to an object on the system unit **3**, to the user working on the web-browser **9** of the client host **10**, the console host **1**, or the control management station **2**. The object pointed at by the web link on the system unit **3** could be an MPEG video, a file (e.g., a text file, PDF, MS power point or MS Word) and etc., and is not limited to these types. It could also be a link of another web service program. From the web browser **9**, the user can directly access the information on the system unit **3** pointed at by the web link without going through the control management station **2** again. In this case, the service modules **8** of the system unit **3** must also include a web server software to directly support the web browser **9** access as mentioned before.



(4) The data path **4** is data flow through a communication link between the system unit **3** and another system unit **3**.

With this path, the service modules **8** in the system units can directly transmit data or information to the service modules **8** of the another system unit via the communication link net**2** **12** without going through the control management station **2**. The communication protocol between the system units **3** could be IP-based or based on any suitable non-IP-based protocol. The data and information transmitted through the path **4** can be in various type such as a data file. For example, a user on the web-browser **9** may navigate through a file folder on the system unit **3**. Later, the user instructs to transfer a file from a current target system unit **3** to another targeted system unit **3** by mouse clicking, therefore, the data file will be transferred directly between two system units **3** without going through the control management station **2** again.



The Variation of the CCDSVM Configuration Models: FIG. **2***a* shows a figure of an example of the CCDSVM with a single control management station **2**. With this model, there is no any system unit **3**, therefore, the CCDSVM degenerated into a single system and the WCUWE just works for the single system, which is the control management station **2**. Thus a user from the web-browser **9** of the console host **1**, a client host **10**, or a control management station **2** can access and operate the entire resources on the control management station **2**.

FIG. **2***b* is a figure of an example of the CCDSVM with identical client host and system unit.

This model is substantially similar to the model of a typical CCSDVM shown in FIG. **1**, except there is no difference between the client host **10** and the system unit **3** since each system unit is also configured with a web-browser **9** and each client host **10** is also configured with service modules **8**. This means that each system unit/client shall play both roles of a system unit and a client host. With this model, a user on each system unit/client can login into the control management **2** from the web-browser **9** and further access and operate the resources of the control management station **2**, or any other system unit/client with a security permission.

FIG. **2***c* is an example of the CCDSVM with non-web accessed client host **10**. This model is substantially similar to the model of an exemplary CCDSVM shown in FIG. **1**, except that the client host **10** has its own service modules **14** for non-web based access instead of using a web-browser for accessing. For example, if the system unit **3** is a SAN unit providing storage volumes, and one of the service modules **14** of the client host **10** is a driver routine for reading data from and writing data to a disk volume on the system unit **3**. The client host **10** may also have a web-browser **9** to access the system unit **3** for other purposes except for accessing disk volumes.

FIG. **3** shows an example of software modules of the WCUWE for a typical CCDSVM.

The software modules of the WCUWE can be implemented with any suitable or a combination of any suitable programming languages such C, C++, Java, JavaScript, Visual Basic, HTML, XML, etc., which means that software of the WCUWE is programming language independent.

In one embodiment, the software of the WCUWE may include following items:

1) web server interface modules **5** of the console support software **6** on the control management station **2**.
2) control management modules **4** of the console support software **6** on the control management station **2**.
3) web server software **7** on the control management station. The web server software **7** could be commercially available web server software on the marketing such as Apache or MS IIS or others, and also could be based on proprietary software. If the web server interface modules **5** provide compatible functionality with the web server software **7**, then the web server software **7** may not be needed.
4) service software modules **8** on the system unit **3**. The service software modules **8** may include web server software **7** or a software module, which provide equivalent functionality as the web server software **7** does.
5) web browser **9** on the console host **1**, on the control management station **2**, or on the client host **10**. The web browser **9** could be a commercial web browser on the market such as MS IE, or Netscape, and also could be proprietary software modules.

FIG. **4** shows simplified examples of a Multi-Layered Item list (MLIL).

FIG. **4***a* illustrates a multi-layered item list contains multiple nodes and multiple layers. A node on an MLIL can be expanded to display sub-node if there is any by mouse clicking. The sub-nodes may be reduced by mouse clicking on the same node again.

FIG. **4***b* illustrates a degenerated Multi-layered items list containing a single node (i.e., a single layer).

FIG. **5** shows simplified examples of operation menu for the WCUWE of the CCDSVM that includes:

FIG. **5***a* illustrates a web based drop-down menu for disk volume management. The drop-down menu can be invoked by right clicking on an item (node) on MLIL.

FIG. **5***b* illustrates a web based selective menu for system unit management. The item selected and further click the submit button.

FIG. **6** shows simplified examples of operation menu for different types of node on an MLIL that includes:

FIG. **6***a* illustrates an operation menu for managing the storage (disk).

FIG. **6***b* illustrates an operation menu for managing the file.

FIG. **6***c* illustrates an operation menu to manage the system unit on network.

FIG. **7** shows an example of system units grouped by a group ID.

According to one embodiment, this figure shows one example of five system units being divided into two groups, a marketing group and an engineer group. The marketing group contains 2 system units while the engineer group contains 3 system units. The groups are automatically and dynamically formed when the system units boot up.

FIG. **8** shows an example of an MLIL combined with mixed devices, system units and system groups That includes:

FIG. **8***a* illustrates an example of a top level of an MLIL is system group, the second level is system unit, the third level is device of disks.

FIG. **8***a* illustrates a top level of an MLIL is system group, the second level is system unit, the third level start with a file folder with files.

The WCUWE can be initiated and established via a sequence of steps, especially these steps are performed by the console support software **6** of the control management station **2** described bellow.

Referring back to FIG. **1**, according to one embodiment, the console support software **6** collects all major resources information of the control management station **2** and converted the information to a presentation in a standard structured format, which is viewable in the web-browser **9**. The resources information mentioned above include the resources of the storage, network, file system, files, folders, users, system status etc, which can be collected whenever the control management station **2** is boot up or at an appropriate time. Such information may be stored in memory for fast retrieve at another time and also may be stored on permanent storage such as disks.

The console support software **6** communicates with the service modules **8** in the system unit **3** to collect all major resources information of the system unit **3** through the communication links provided by the net**2**. Further, the console support software **6** binds and organizes the collected information of the system units **3** as part of network information of the control management station **2**. In addition, upon the access from the user, all the collected network information is organized and converted to a standard structured format for web communication, and subsequently is viewable in the web-browser **9**. The collected network information includes the information of the system units **3** and their storage device, network devices, wireless devices, video and audio record & playback devices, file-system, users, system status and all others on net. The network information can be collected whenever the system units **3** is boot up or at an appropriate time.

Further, the network information of the control management station **2** can be kept in memory for later fast access and can be organized into any suitable logical structure, for example as a simple list of array, a linked list, a double linked list, a hash table, a tree structure and etc. The network information list can also be stored on permanent storage such as disk or others for the backup purpose. In addition, the network information can be stored in any form on permanent storage.

An example of the form may be a commercial database, a binary record file, a flat text ASCII file or others. All specific examples of a form for a data depository or a logical structure mentioned above are for illustration only, and they shall not be limited to these mentioned herein in this invention.

After the initiation of the WCUWE, each of the users via an associated computing system (device) through a web-browsers **9** is allowed to login the CCSVM, thus various resources & information of the CCDSVM provided by the WCUWE can be displayed, viewed, accessed, or operated.

Users Interaction, Access and Manage Resources with the WCUWE: Users from the client host **10**, the console host **1**, or the control management station **2** can log into the WCUWE of the CCSDVM via a web URL link provided by the software of the WCUWE (see FIG. **3**). The software of the WCUWE presents information and resources on the control management station **2** and on the server units **3** to the users after their login from their own web browsers **9**. Subsequently, whenever a user requests to access or operate a major resource either on the control management system **2** or the system unit **3** of the CCDSVM, the software modules of the WCUWE is configured to create a corresponding user task for such a request on the targeted system. In one embodiment, the software of the WCUWE is configured to support multiple simultaneous tasks for the same user from a single web-browser **9** on an associated computing device; and also to support multiple concurrent users each from his or her own web-browser on an associate computing device to perform such tasks while providing best security protection for the resources being accessed by the multiple users.

In order to provide convenience for each of the login users to perform tasks in the WCUWE, the console support software **6** may use a logical structure of multi-layer item list (MLIL) as illustrated in FIG. **4** together with the associated web-browser based operation menu as illustrated in FIG. **5** for operating corresponding actual multi-level organized resources. The resources as mentioned before could be the file folders of a file system, the system units **3** on the network, the users and their security profile etc. If there is a need to create a structured multi-layer item list (MLIL) of FIG. **4** for resources of the CCDSCM upon each login user trying to access and operate, the console support software modules **6** of the control management station **2** are configured to do the following:

a) For each of log on sessions of each of users, a memory management module of the console support software **6** is configured to create an MLIL which can be flexibly expanded or reduced by mouse clicking thereon while being displayed in a web browser, where the created MLIL mirrors an actual resource structure. Further, the console support software **6** makes an association between each created MLIL and the actual targeted resources. The created MLIL can reside in memory and can be in any form of a list structure such as a simple array list, link list, double link list, various tree structure, or hashed table, etc.

b) For each created MLIL, binding a corresponding graphic or text representation to each node (item) on the MLIL for the purpose of displaying, and associating each node (item) with appropriate attributes. These attributes shall reflect characteristics of the actual resources, such as name, type, level, size etc. and of course it is not limited to these, therefore, the actual resources can be correctly displayed through the MLIL.

c) Link a corresponding operation menu to each node of the MLIL. The operation menu can be either a web-based drop-down menu **1** as shown in FIG. **5** or a non-drop-down selective menu **2** as shown in FIG. **5**.

d) The created MLIL and all its associated information are converted into the standard structured information (web page) for web communication, such that to be viewable in and accessible via the web-browser **9**. Thereafter, the MLIL can be either viewed in a web-browser **9** on the control management station **2** or viewed by a web-browser **9** on the client host **10** or the console host **1**. The converted logical item list needs to be transmitted via a communication link net**1** **11** or net**2** **12** from the control management station **2** to the client host **10** or the console host **1**. The converted logical item list can be transmitted using a web protocol such as HTTP or any other suitable protocols in order for users remotely view it from the web-browser **9**.

e) After user taking action of either viewing, or operating on any of resources represented by the MLIL, the console support software **6** modifies the MLIL through the memory management module of the console support software **6**. Also, the memory management module of the console support software **6** keeps tracks of the memory usage and status for the correspondent MLIL as the results of expanding or reducing the MLIL when a user accesses or operates a resource represented by this MLIL. For example, deleting a file folder operation requires to delete an actual folder on a file system and delete the corresponding node on the MLIL. Most importantly, all processes or threads created to perform tasks of operating on the resources represented by the MLIL can effectively share the dynamically modified same MLIL with a consistent view throughout the entire login session of a login user. The memory management module of the console support software **6** may need to translate and map the initial logical memory address of the MLIL to another logical memory address, therefore, the different processes or threads of the same user session can effectively access the same actual MLIL memory object during the user session,

f) The operations described from b) to e) above can be repeatedly processed as long as a user keeps working on the same resource represented by this specific MLIL.

For each of logout users, the created MLIL for the user session will be deleted through the memory management module of the console support software **6**. Further if a user stops accessing the current related resource and requests to access a total different resource, the original MLIL will be deleted and a new MLIL will be created.

The Support of Deeply Nested Files & Folders Lists on Systems within the WCUWE of the CCDSVM:

The accessing and managing a file system always has been an important part of a computer user work environment (CUWE) in the past. Supporting a file system to be accessed by users from a network has always being a challenge to many vendors. With the present invention, a user can manage storage volumes of the entire CCDSVM from a web browser **9** and further can create, manage, and access the file system on either the control management station **2** or the system unit **3**. With a sophisticate method of using the multi-layer item list as shown FIG. **4** together with the operation menu shown in FIG. **5** for the file system in the WCUWE of the CCDSVM in this invention, users can efficiently access and manage files and folders in file systems from a web-browser **9** of an associated computing system on the fly without caching anything for the file system.

In one embodiment, the multi-layer item lists (MLIL) being used to represent an actual file folder structure of a file system on a target system of the CCDSVM has each node (item) to represent a folder or a file system or a file. When a user navigates through the file system and uses mouse-clicking on the MLIL to operate a file or folder from a web-browser **9**, only a small piece of information, which is related to the file

or folder, needs to be retrieved or transmitted between a target system of the CCDSVM and the control management station **2**. For example, if user wants to add or delete an item such as a folder, the only major information needs to be transmitted is a folder name from the control management system **2** to the target system and the target system shall carry out the task. If a user wants to access the information, which is bellow a current item on the MLIL, such as a folder, after sending a folder name, the only information that needs to be retrieved from the target system is the sub-folder or file names if there is any under the current folder. If the user wants to set access permission for a file or a folder on any of target systems for one or more designated users to access, the information of the targeted system and information of the file or folder on the network need to be bound with each designated user's security profile. The target systems could be either the control management station **2** or any of the system units **3** in the CCDSVM.

If a user needs to view or edit the content of a file on a system unit **3**, the file may be transmitted via the communication link net **1** & net **2** directly from the target system to the client host **10** or the console host **1**, without going through the control management station **2**. If the file is on the control management station **2** and users like to view or edit it from a remote system, the file is transmitted via the communication net **1** to the client host **10** or the console host **1**. With a proper viewing tool on the client host **10** or the console host **1** or the control management station **2**, which may be invoked from the web-browser **9**, the user is able to view the contents of the file. With a proper editor, user can further edit the file and then transmit the file back to the target system. The target system could be either the system unit **3** or the control management station **2**. Alternatively, the file may also be transmitted from the target system to the control management station **2** and then convert it from original format to a format, which may be viewed and edited in the web-browser **9**. After converting, if the target system is not the control management station **2**, the file can be transmitted via the communication link net **1** or net **2** from the control management station **2** to the client-host **10**, the console host **1** for viewing or editing with a web-browser **9**. If the target system is the control management station **2** itself, there would be no such transmission required.

The Support of the Automatically & Dynamically Provisioned Groups of Systems in the WCUWE of the CCDSVM:

Network information is an important part of a WCUWE. In one embodiment, the server unit **3** of the CCDSVM can be automatically and dynamically provisioned into multiple groups based on their group ID, where each group contains at least one server unit **3** ("provisioned system").

To effectively reflect such nature of the CCDSVM, and further to allow user to access and manage such grouped system units **3** and their associated devices, such as storage or network interface cards or terminal monitor or video & audio recorder (e.g., camcorder, or keyboard & mouse), or wireless devices or file systems on the network, the multi-layer item list (MLIL) can be used. Therefore, in one embodiment, the mixed information on the network can be ordered with a top-down fashion such that the server-group at top level, the system units **3** at the second level, and the hardware devices or file system start from the third level and so on with the MLIL.

More specifically, a node on the MLIL with different levels or types may associate with a different type of an operation menu. For example, a node of the system unit **3** on an MLIL may be associated with an operation menu containing system status, shutdown, storage management, change usage etc. and a node of storage device may be associated with an operation menu of creating a storage volume, displaying storage volume, and delete a storage volume and so on. After converting all information associated with the MLIL to a standard structure, the complex information on the network can be viewed and operated by a user from a web-browser **9**.

For example, FIG. **7** shows the example of how each of the system units **3** is provisioned into multiple groups that is represented by an MLIL. FIG. **8** shows the example of how does devices such as storage being listed under the server units **3** in each group. FIG. **9** shows an example of how the file system on a system unit **3** in a group can be accessed. As described herein, one embodiment of automatically and dynamically grouping each of the system units **3** and wisely using the MLIL to represent the diversified resources on the network, the entire CCDSVM can be much more efficiently accessed, operated, and managed. For example, through mouse clicks on a web-browser, privileged users can create storage volumes from a fresh disk on any target system and make a file system on each of the storage volumes, mount the file system and create a folder structure on the file system, and further setup an access control for files and folders of the file system for users with limited privileges to access. Therefore, the WCUWE can securely let permitted user access from a web-browser anywhere on the network.

The Support of the User Administration & Authentication in a WCUWE of the CCDSVM:

Initially, a WCUWE allows a default super user to login the WCUWE of the CCDSVM and to access an authentication web-page provided by the console support software **6**. This privileged user can setup other privileged or non-privileged users account and their respective profiles thereafter. Similar to a file system in the WCUWE of the CCDSVM, a user profile can be viewed and operated with using an MLIL combined with a web-based operation menu as shown in FIG. **5**, and further to convert it to a standard structure, which can be viewed and accessed by the user from a web-browser **9**. Therefore, through mouse click on the MLIL pertaining to the user profiles and a web-based operation menu in a web page, the user can perform operations of user profile creating, viewing, updating, deleting and others without limits. Specially, the privileged users can setup profiles for other non-privileged users for the secure access control. In addition, the user information and their security profiles can be kept in any form in a database, which could be a commercial database on the market, or a plaint text file, or a binary record file, or others.

The present invention has been described in considerable details with preference to certain preferred versions, examples, and figures; however, other versions, and samples are also possible. Therefore, the spirit and scope of the appended claims should not be limited to the description of the preferred versions and samples contained herein.

What is claimed is:

1. A server supporting a plurality of users access to remote folder structures, the server comprising:

memory, and non-transitory computer-readable medium comprising

program code which, being executed by the server, configures the server to:

create a first per user-session hierarchical list in the memory for a user session initiated via a first end-user device by a first one of the users for access to a folder structure served by the server, the first hierarchical list representing the folder structure in a reduced form, the folder structure comprising one or more folders, where each of the one or more folders is used for holding at least one data object,

send a user interface comprising the first hierarchical list to the first end-user device to be displayed thereon, the

displayed first hierarchical list being navigated by the first one of the users to request access to the folder structure;

process the request for access to the folder structure received from the first end-user device, wherein the program code to process the request includes to update the folder structure, and also update the first hierarchical list in the memory to reflect the updated folder structure in accordance to the request,

wherein the server sends an updated user interface comprising the updated first hierarchical list to the first end-user device to be displayed thereon during the user session, and deletes the first hierarchical list from the memory in response to exit of the user session.

2. The server as recited in claim **1**, wherein the server

creating a second per user session hierarchical list in the memory to represent a folder structure in a reduced form in response to a user session initiated via a second end-user device by a second one of the users, allowing the second one of the users via the second hierarchical list displayed on the second end-user device access to the folder structure served by the server.

3. The server as recited in claim **1**, wherein a folder structure served by the server is stored in a storage space which is configured in a storage device, wherein the server controls access to one or more storage devices, each of the one or more storage devices being configured with one or more storage spaces.

4. The server as recited in claim **1**, wherein said data object further is one of a file, or a folder.

5. The server as recited in claim **1**, wherein said data object further is a message.

6. A method for supporting a user access to remote folder structures, the method comprising:

creating, by a server, a per user-session hierarchical list in a memory for a user session initiated via an end-user device by the user for access to a folder structure served by the server, the hierarchical list representing the folder structure in a reduced form, the folder structure comprises one or more folders, where each of the one or more folders is used for holding at least one data object,

sending a user interface comprising the hierarchical list to the end-user device to be displayed thereon, the displayed hierarchical list being navigated by the user to request access to the folder structure;

processing the request for access to the folder structure received from the end-user device, the processing of the request including to update the folder structure, and also update the hierarchical list in the memory to reflect the updated folder structure in accordance to the request,

wherein the server sends an updated user interface comprising the updated hierarchical list to the end-user device to be displayed thereon during the user session, and deletes the hierarchical list from the memory in response to exit of the user session.

7. The method as recited in claim **6**, wherein said navigating the hierarchical list to request for access to the folder structure further comprises:

moving a data object into one of the one or more folders in the folder structure or removing a data object from one of the one or more folders in the folder structure.

8. The method as recited in claim **6**, wherein said navigating the hierarchical list to request for access to the folder structure further comprises:

sharing a data object in the folder structure with at least a designated user.

9. The method as recited in claim **8** further comprises:

causing the server to add information of the data object into a security profile relating to the designated user for permitting the designated user access to the data object; and further comprises:

validating the designated user for access to the data object in respect to the information of the data object specified in the security profile.

10. The method as recited in claim **6**, wherein said navigating the hierarchical list to request for access to the folder structure further comprises:

renaming or copy/paste a data object in one of the one or more folders in the folder structure.

11. The method as recited in claim **6**, wherein the hierarchical list is expandable to be explored with mouse-like clicking by the user during a user session.

12. A computer program product, for supporting a plurality of users access to remote folder structures, tangibly stored in a non-transitory computer-readable medium in a server, the computer program product comprising:

program instructions to be executed by the server for the server to:

create a first per user-session hierarchical list in a memory for a user session initiated via a first end-user device by a first one of the users for access to a folder structure served by the server, the first hierarchical list representing the folder structure in a reduced form which comprises one or more folders, where each of the one or more folders is used for holding at least one data object;

send a user interface comprising the first hierarchical list to the first end-user device to be displayed thereon, the displayed first hierarchical list being navigated by the first one of the users to request access to the folder structure; and

process the request for access to the folder structure received from the first end-user device, wherein the program instructions to process the request includes to update the folder structure, and also update the first hierarchical list in the memory to reflect the updated folder structure in accordance to the request,

wherein the server sends an updated user interface comprising the updated first hierarchical list to the first end-user device to be displayed thereon during the user session, and deletes the first hierarchical list from the memory in response to exit of the user session.

13. The program product as recited in claim **12** comprising program instructions for the server

creating a second per user session hierarchical list in the memory to represent a folder structure in a reduced form in response to a user session initiated via a second end-user device by a second one of the users, allowing the second one of the users via the second hierarchical list displayed on the second end-user device access to the folder structure served by the server.

14. The program product as recited in claim **12** further comprising program instructions for allowing via a hierarchical list to move data object into one of the one or more folders in the folder structure or remove data object from one of the one or more folders in the folder structure.

15. The program product as recited in claim **12** further comprising program instructions for allowing via a hierarchical list to share data object in the folder structure with at least a designated user.

16. The program product as recited in claim **15** further comprising program instructions for causing the server to add

information of the data object into a security profile relating to the designated user for permitting the designated user access to the data object; and

further comprises program instructions for validating the designated user for access to the data object in respect to the information of the data object specified in the security profile.

17. The program product as recited in claim **15** further comprising program instructions for allowing via a hierarchical list to

rename or copy/paste a data object in one of the one or more folders in the folder structure.

18. The program product as recited in claim **15** further comprising program instructions for allowing the hierarchical list to expandable to be explored with mouse-like clicking by a user during a user session.

19. The program product as recited in claim **15** further comprising program instructions for the server controlling to configure each of a plurality of storage devices to comprising one or more storage spaces, where each of the one or more storage spaces is configured with a folder structure.

20. The program product as recited in claim **15**, wherein said data object further is one of a file, or a folder or a message.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 9,280,547 B2
APPLICATION NO. : 13/914515
DATED : March 8, 2016
INVENTOR(S) : Tsao

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Title Page, please insert, item (63)

--Related US Application Data

continuation of application No. 13/050,871, filed on 03/17/2011 and now a Pat. No. 8589527, which is a continuation of application No. 11/374,302, filed on 07/02/2004 and now Pat. No. 7945652;

said application No. 11/374,302 is a continuation-in-part of application No. 10/713,905, filed on 8/12/2002, now Pat. No. 7379990, and is a continuation-in-part of application No. 10/713,904, file on 8/6/2002, now Pat. No. 7418702.--

Signed and Sealed this
Twenty-eighth Day of June, 2016

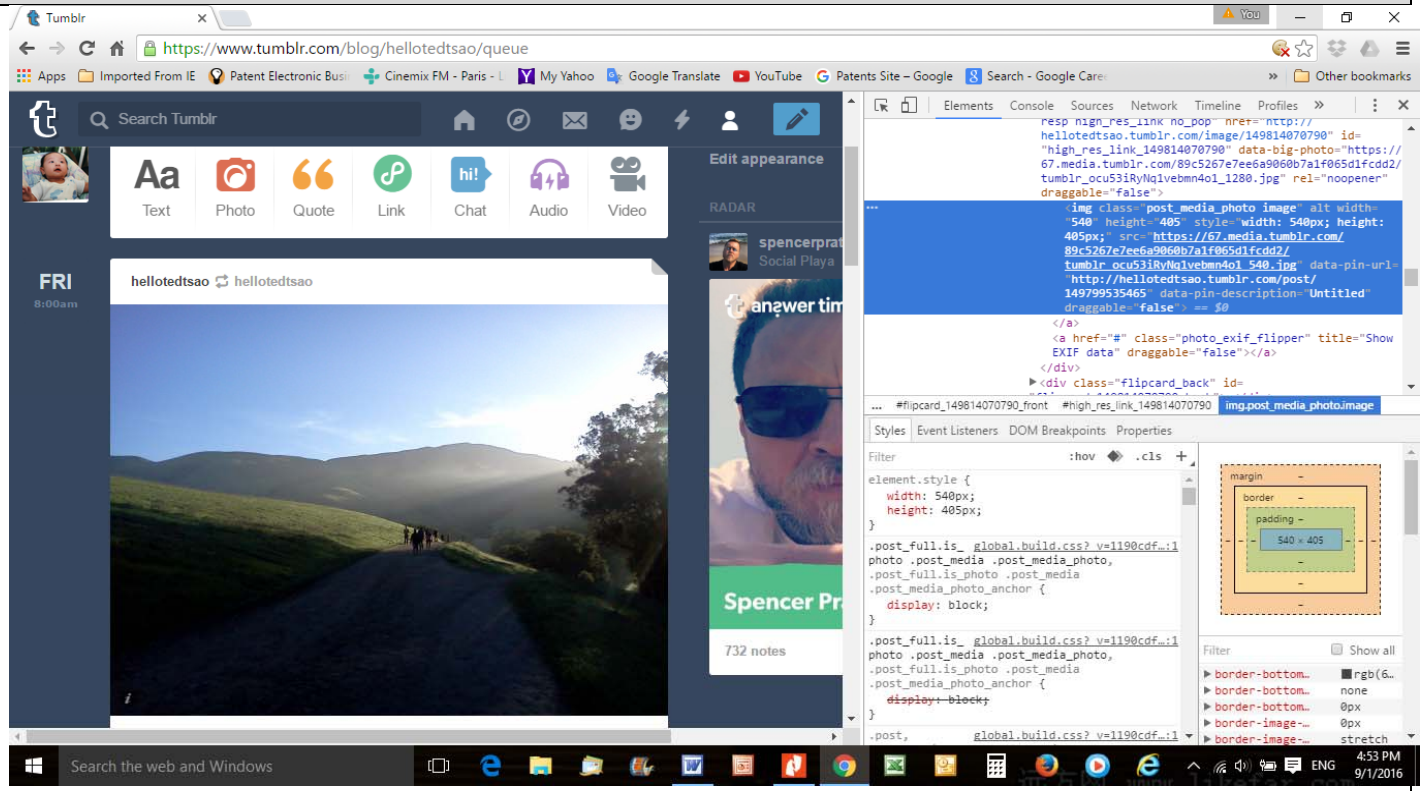
Michelle K. Lee

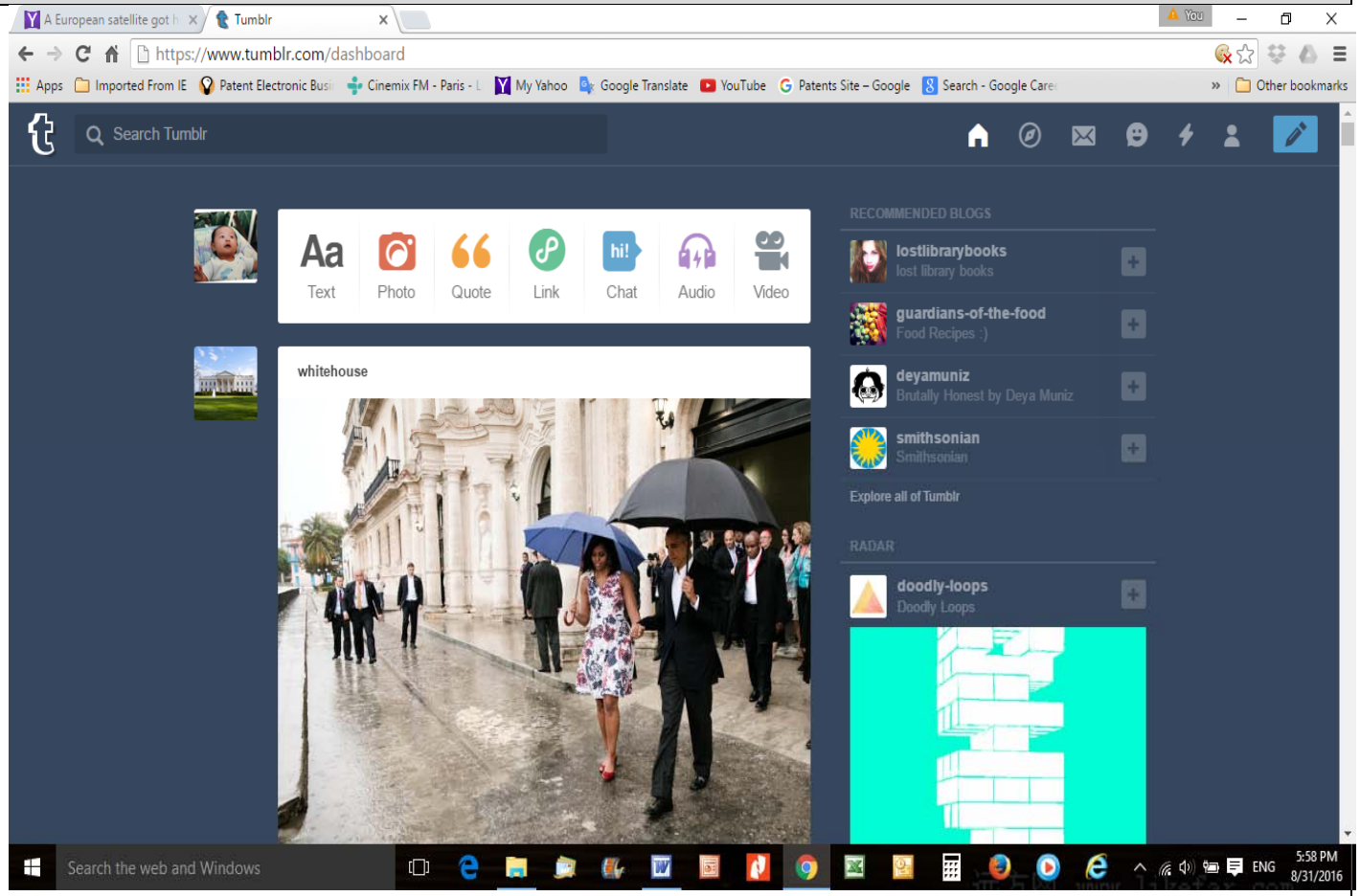
Michelle K. Lee
Director of the United States Patent and Trademark Office

Exhibit F

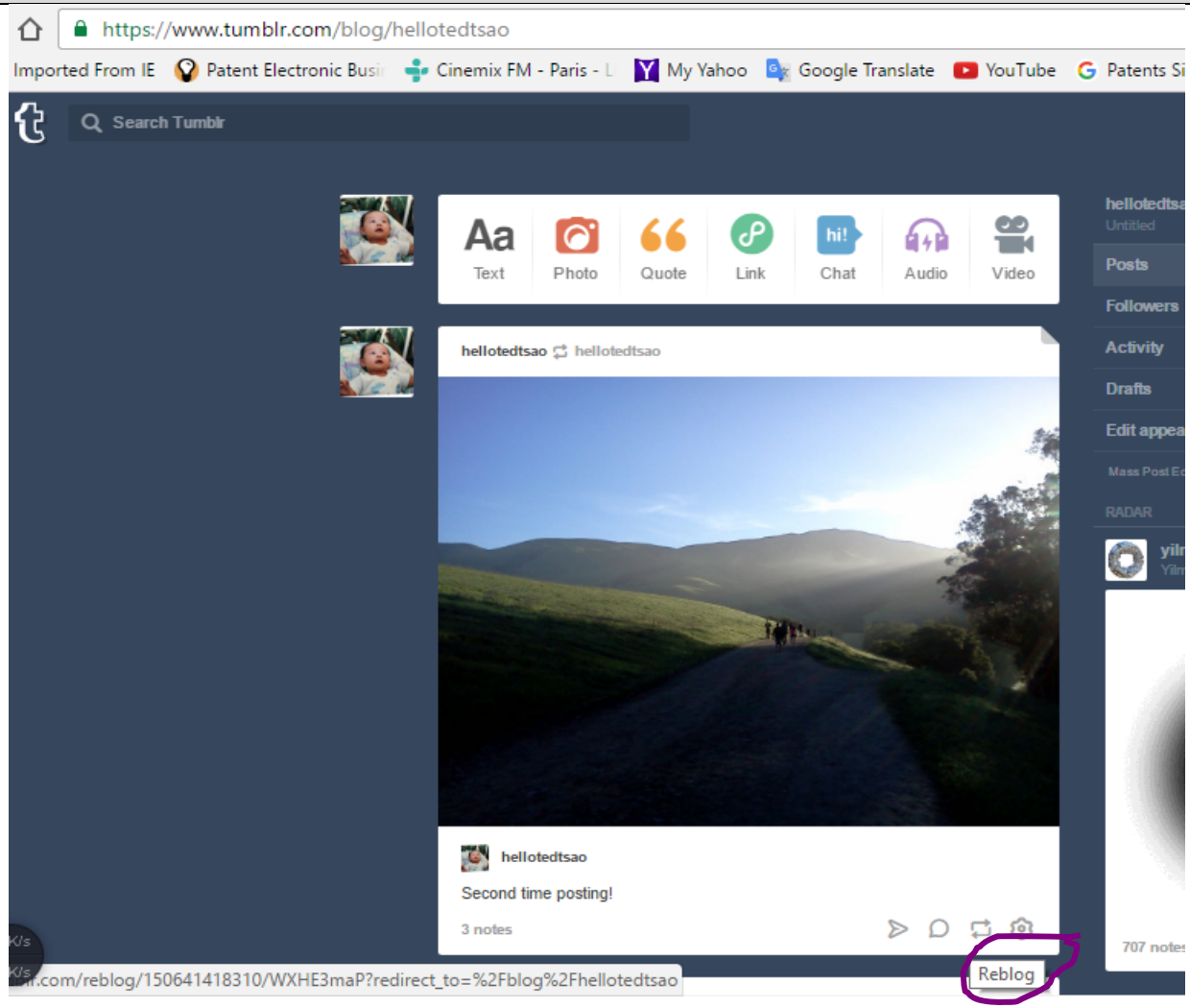
Illustrative Claim Chart 1
U.S. Pat. No. 8,396,891

Claim 1	Yahoo's Tumblr Service ("Tumblr")
1. A computing device comprising a processor, memory and program code which, when executed by the processor, configures the device to:	Tumblr's server is a computing device that has a processor, memory and program code. The program code, when executed by the processor, causes the server to perform the following functions:
(i) display a user interface to each of a first user and a second user to share information, wherein each of the user interfaces comprises, for each of the first and second users,	(i) To display a user interface to at least two different users. Two users' Tumblr accounts were used for the test, "hellotedtsao" and "bethannyt." The user Bethannyt is a follower of the user hellotedtsao. The interface of each of the two user accounts (or any Tumblr user's account) contains the following, to use hellotedtsao as an example:
(a) a private section configured to display information about files or folders available for the user to share and	(a) hellotedtsao's private section. The private section is displayed via https://www.tumblr.com/blog/hellotedtsao, and is configured to display information about a mountain trail photo file in the screenshot below. The photo file physically resides on a computer located at 67.media.tumblr.com. The photo file is normally not viewable by others, but is available for the user to share with other Tumblr users if he or she so chooses.

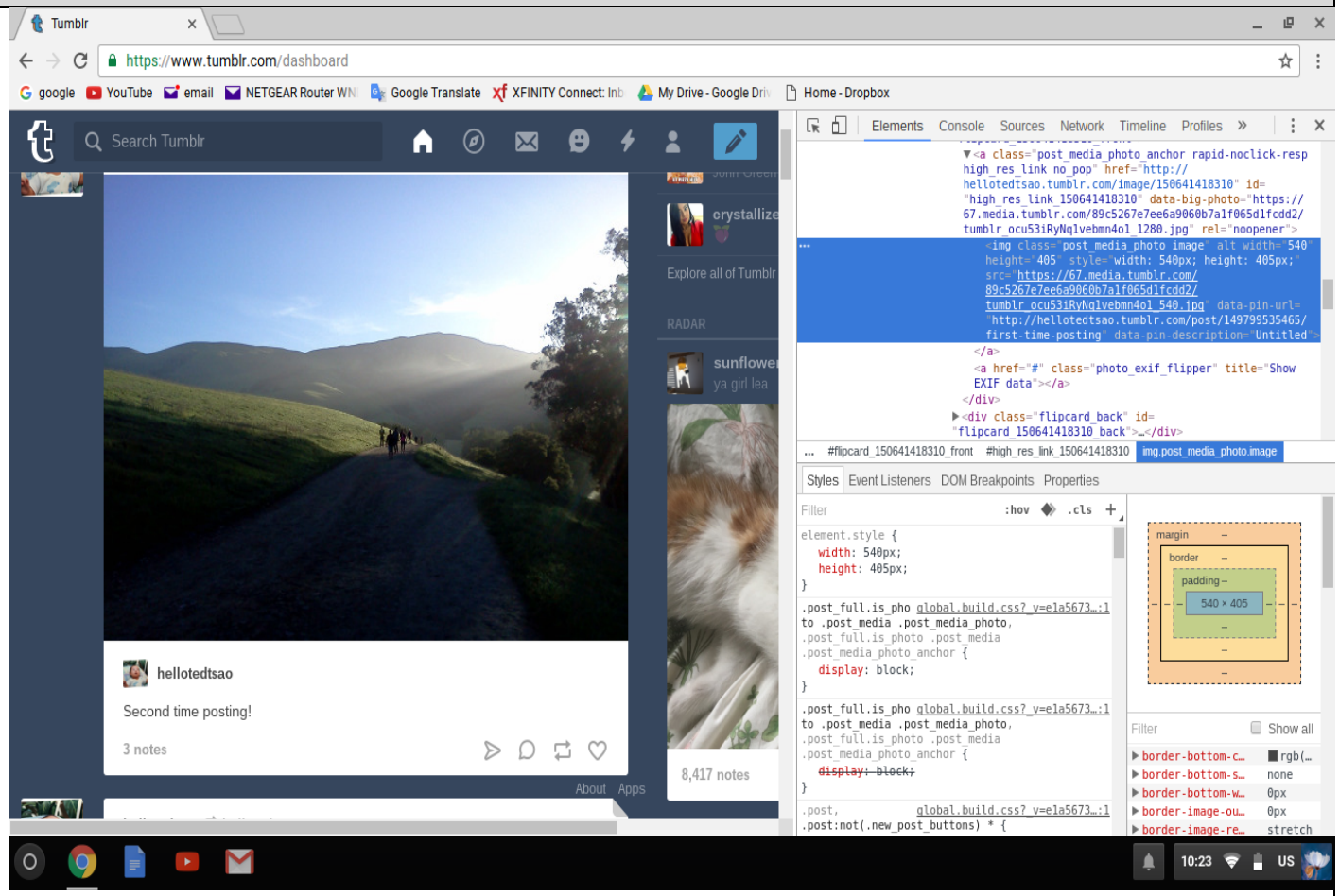
Claim 1	Yahoo's Tumblr Service ("Tumblr")
	
(b) a common section configured to display information about files or folders shared with the user;	(b) hellotedtsao's common section. The common section is displayed via "https://www.tumblr.com/dashboard as shown in the screenshot below. The common section is configured to display information about files that hellotedtsao has shared with other users.

Claim 1	Yahoo's Tumblr Service ("Tumblr")
	
(ii) share a file or folder selected, from the available files or folders, by the first user with the second user by	(ii) To share a selected file or folder with another user, bethannyt, by the following means:

Claim 1	Yahoo's Tumblr Service ("Tumblr")
(a) allowing the first user to identify the file or folder in the private section on the first user's interface, which is not viewable by the second user,	(a) allowing the user hellotedtsao to identify the photo file in the user's private section on user's web browser interface. This photo files was not visible to the other user bethannyt until the user hellotedtsao selected the photo file and click the "Reblog" button, which allows all of hellotedtsao's followers, including bethannyt, to view the selected file.

Claim 1	Yahoo's Tumblr Service ("Tumblr")
	
(b) unlocking a protection mechanism of the file or	(b) unlocking a privacy protection mechanism of the photo file, after the user hellotedtsao presses the "reblog" button. The privacy protection mechanism prevents all the files in the private section from

Claim 1	Yahoo's Tumblr Service ("Tumblr")
folder to allow access to the second user,	being seen by other users. Unlocking the privacy protection mechanism allows the other user bethannyt to access the photo file.
(c) storing information about the file or folder, without the content of the file or folder, in a common work place accessible to both the first user and the second user, and	(c) Storing information about the photo file, without the content of the photo file, in a common work place accessible to both the first user and the second user. This is evidenced by the fact that the identified photo file, which always physically resides on the computer located at 67.media.tumblr.com, can be "reblogged" multiple times to share with multiple users who are followers of the user hellotedtsao. Essentially, such stored information is similar to a "pointer" that points to the identified photo file, instead of moving the file itself. The screenshot below shows that after hellotedtsao reblogged the photo file, it still physically resides at 57.media.tumblr.com.

Claim 1	Yahoo's Tumblr Service ("Tumblr")
	
(d) displaying information about the file or folder in the common section on the second user's interface, wherein the second user can	(d) displaying information about the photo file in the common section the user bethannyt's browser interface at "https://www.tumblr.com/dashboard." User bethannyt can now access the photo file through the displayed information. The screenshot below bethannyt's user interface displaying the mountain trail photo posted by the user hellotedtsao.

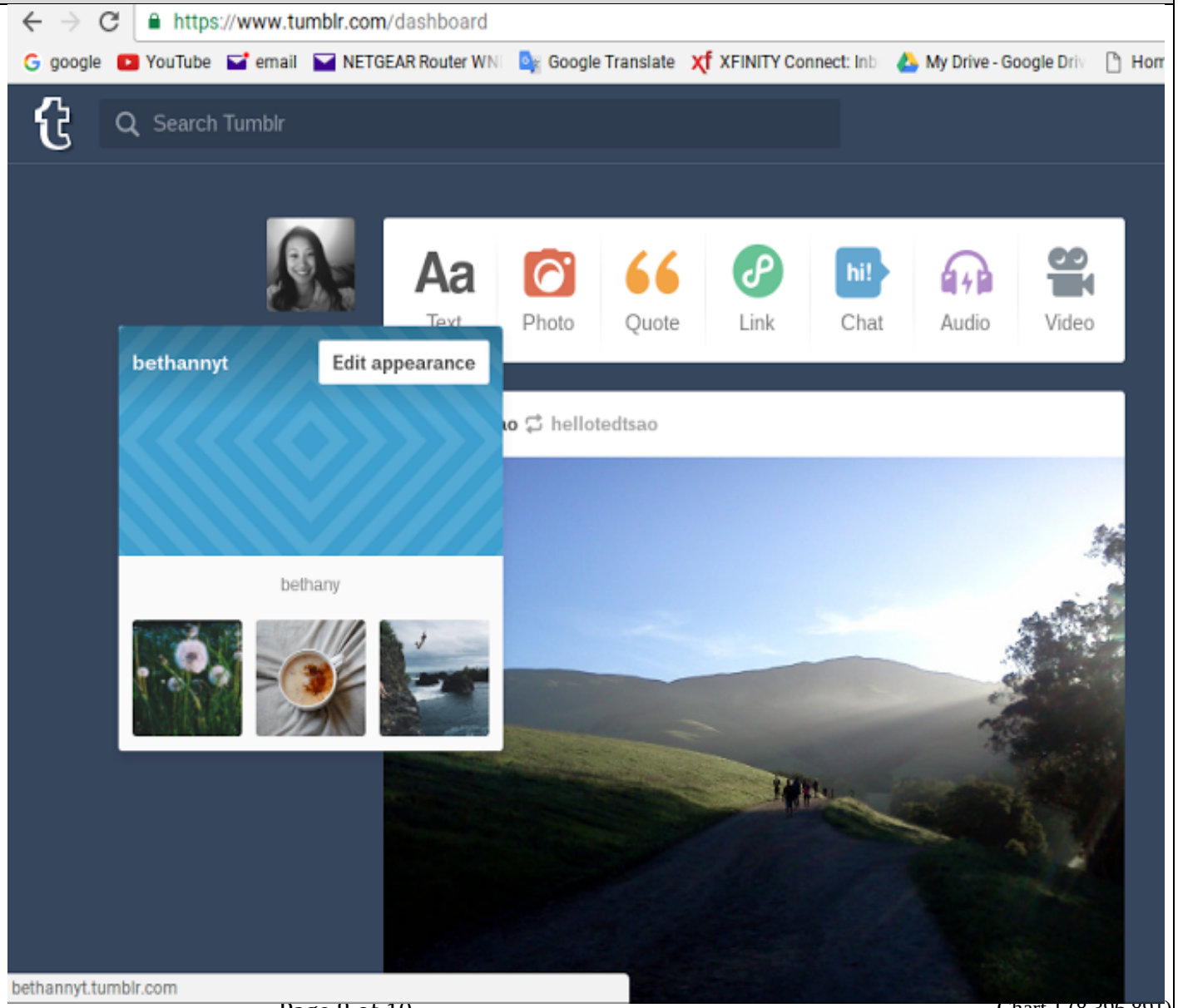
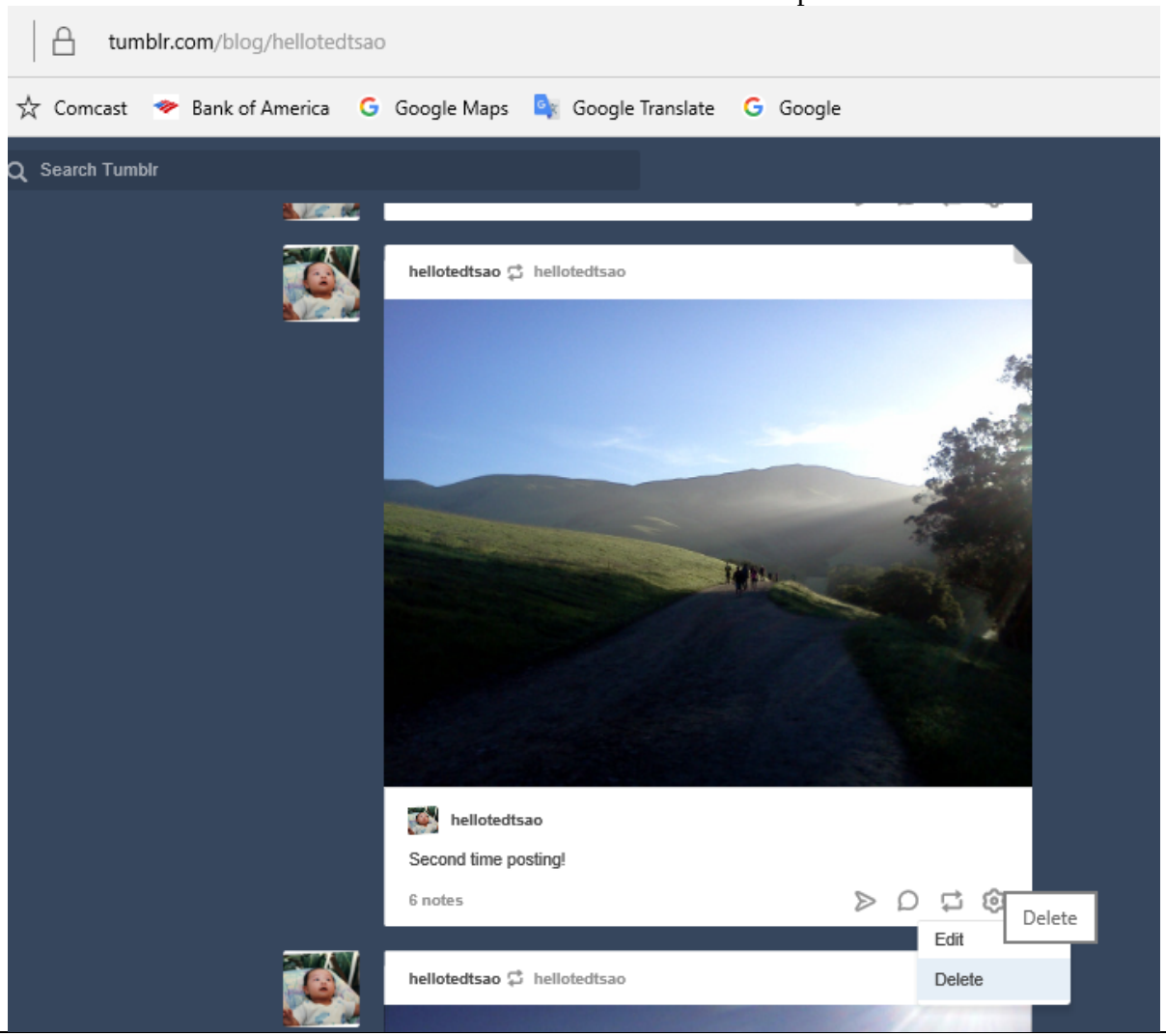
Claim 1	Yahoo's Tumblr Service ("Tumblr")
access the file or folder through the displayed information; and	

Chart 1 (8,396,891)

Claim 1	Yahoo's Tumblr Service ("Tumblr")
(iii) stop sharing of a file or folder to the second user that the first user has previously shared with the second user by	(iii) To stop sharing of the photo file with the user bethannyt, the user hellotedtsao may select the photo file and click the "delete" button either from his private section or from the common section. The screenshot below shows the "Delete" function in hellotedtsao's private section. 

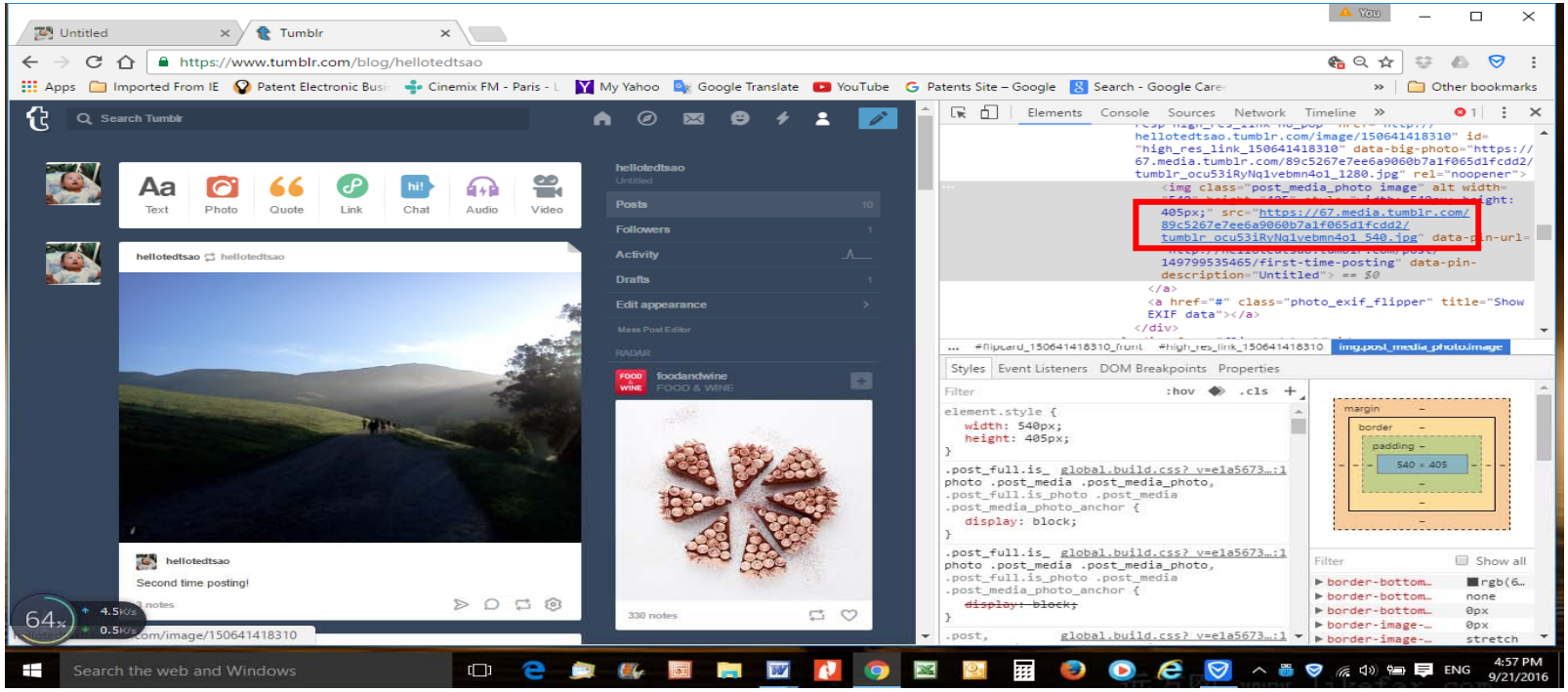
Claim 1	Yahoo's Tumblr Service ("Tumblr")
(a) deleting information about the file or folder displayed in the common section on the second user's interface,	(a) when the user removes a post from the common section, by pressing the "Delete" button, Tumblr's server deletes the information about the photo file displayed in the common section on the the user's followers' interface.
(b) deleting information about the file or folder that has been stored in the common work place, and	(b) Tumblr's server deletes information about the photo file that has been stored in the common work place, consequently the photo file becomes inaccessible from the common section.
(c) locking the protection mechanism to rescind access to the second user.	(c) Tumblr's server also replaces the privacy protection mechanism to make the photo only accessible from hellotedtsao's private section.

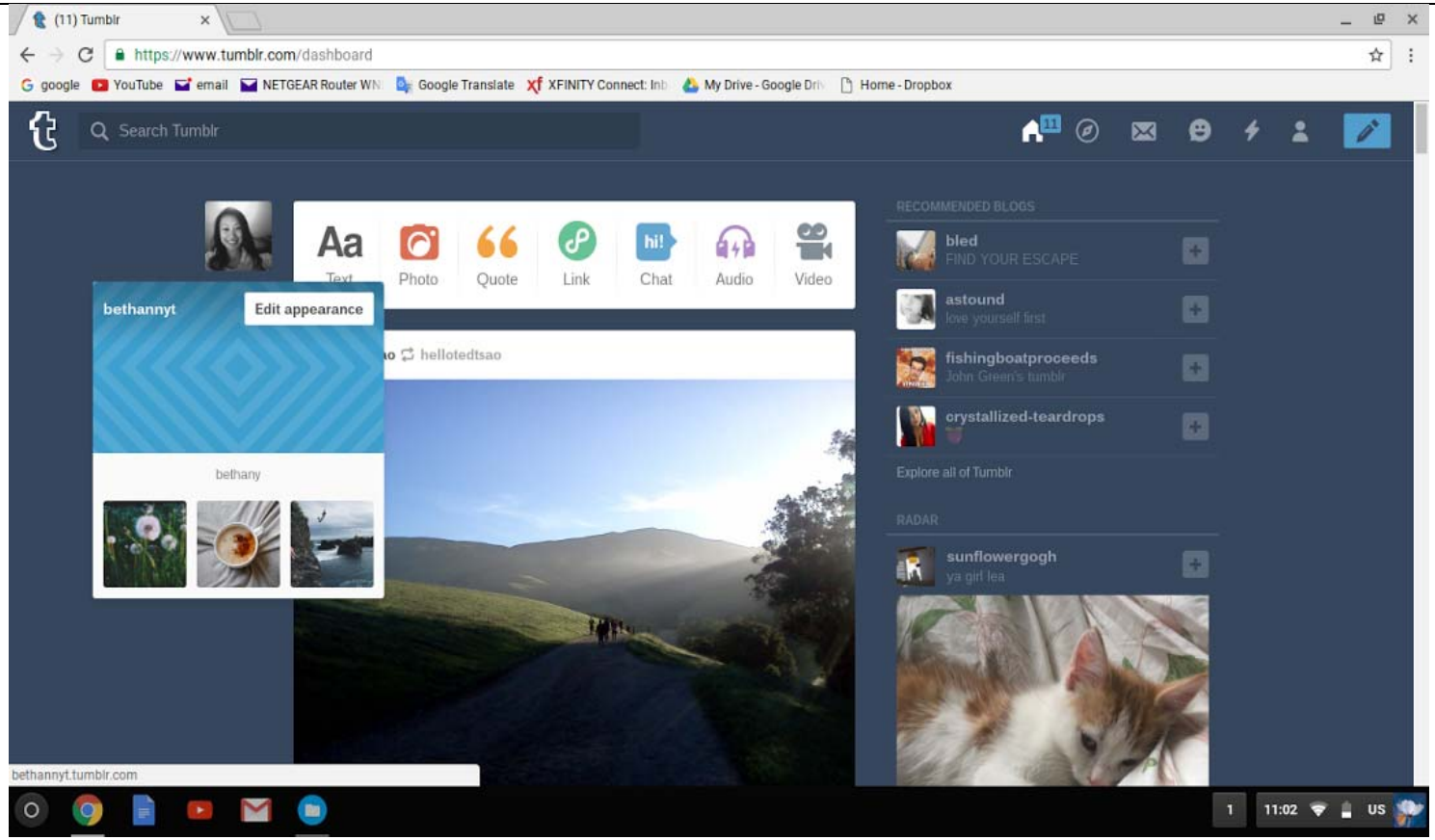
Exhibit G

Illustrative Claim Chart 2
U.S. Pat. No. 8,713,442

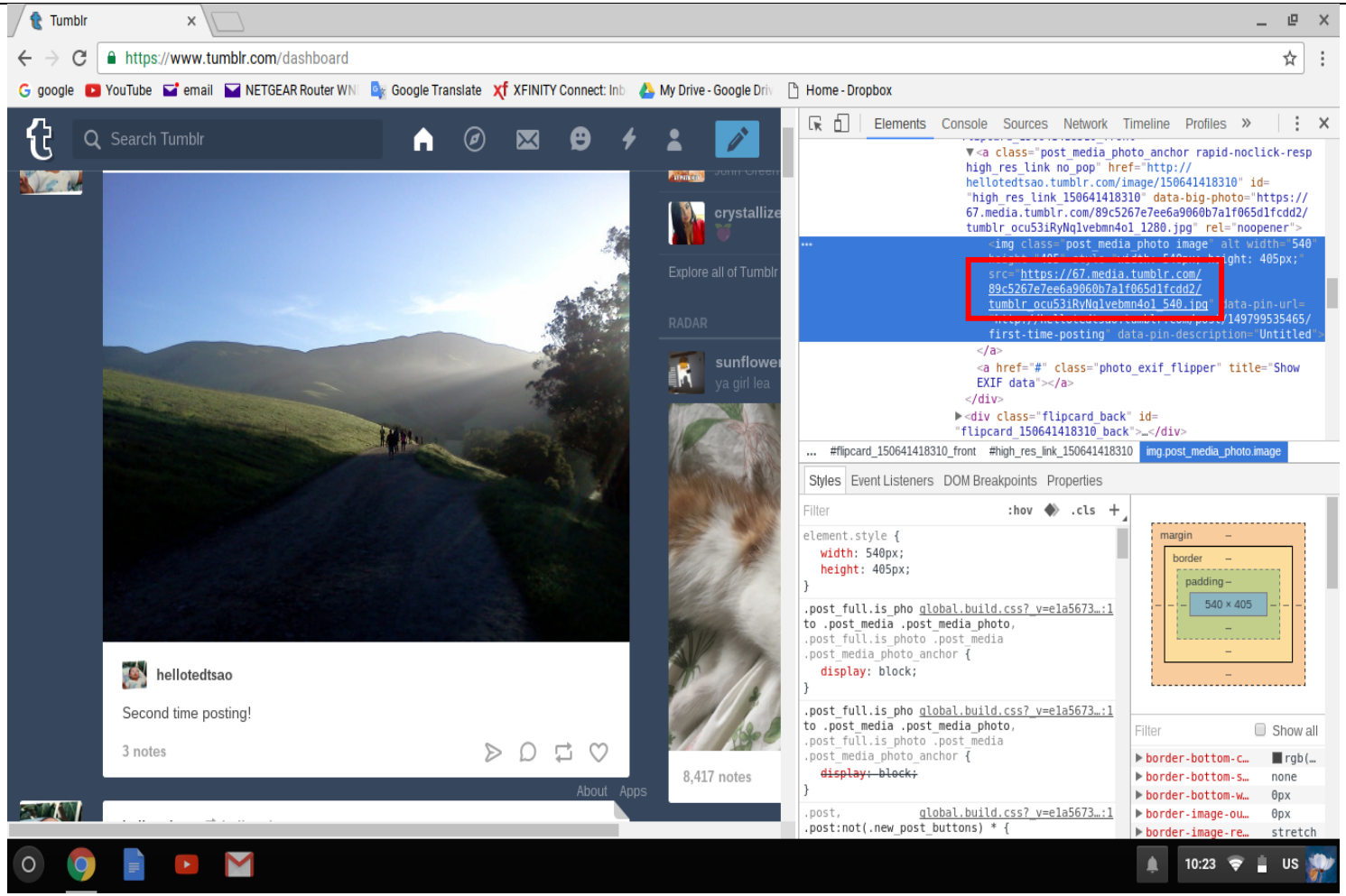
Claim 9	Yahoo's Tumblr Services ("Tumblr")
9. A server in a collaboration system supporting virtual presentation, the server comprising:	The server or servers for Tumblr (collectively the "Server") allow online users to make virtual presentation to the user's followers through the "blog" and "reblog" features.
at least one hardware processor, one network interface device, and program code, in a storage medium, which when executed by the at least one hardware processor causes the server to:	The Server has at least one hardware processor, one network interface device, and program code in a storage medium, which when executed by the processor or processors causes the server to perform the following operations:

cause displaying a first user interface comprising metadata of files and folders, residing in the server or in at least one computing device, on a first end-user device to allow a first user selecting one file or one folder from the metadata displayed and requesting the metadata of the selected file or folder to be posted to a second user interface;	When a user wants to blog, e.g., a picture on Tumblr, the user uploads the picture onto Tumblr, which causes the user's device to display the metadata of the photo, which may include comments, notes, graphical indicators, a thumbnail image or even a low resolution copy of the picture. These metadata reside on the server and also on the user's device. Many such uploads can be added to a queue. These metadata in the queue can be displayed on the user's device to allow the user to pick one picture and hit the POST button to post the picture to a user interface of his or her follower who is the "second user" within the meaning of the claim language. The posting can be a broadcast to all followers or a private post to a selected follower. The Reblog button of Tumblr meets this element in a similar way.
store the metadata, but not content, of the selected	In processing the Post or Reblog request, the Tumblr Server stores the metadata of the blogged photo onto the follower's user interface, without storing or moving the picture itself. This is evidenced by the following test: Before a mountain road picture was reblogged by a first user (hellotedtsao), the picture is shown to reside on a particular folder on 67.media.tumblr.com. After the reblogging, the second user (bethannyt) who is a follower of the first user, received all the metadata of the blogged picture on her user interface. But on the second user's dashboard, the location of the picture file is still on the same folder at 67.media.tumblr.com. This test is shown by the

file or folder according to the request for the posting received from the first end-user device; and	following three screen shots: (1) Screen Shot 1: Before the Reblog as viewed from the first user interface showing the mountain road picture to be reblogged and the folder location of the jpg file of the picture as indicated by the red rectangular box.  (2) Screen Shot 2: After the Reblog, as viewed from the second user interface, with the all metadata of the reblogged mountain road picture copied onto a follower bethannyt's user interface.



(3) Screen Shot 3: After the Reblog as viewed from bethannyt's user interface showing the folder location of the reblogged picture, which is identical as the folder location of the picture as viewed from the first user interface before the reblog in Screen Shot 1 as indicated by the red rectangular box.

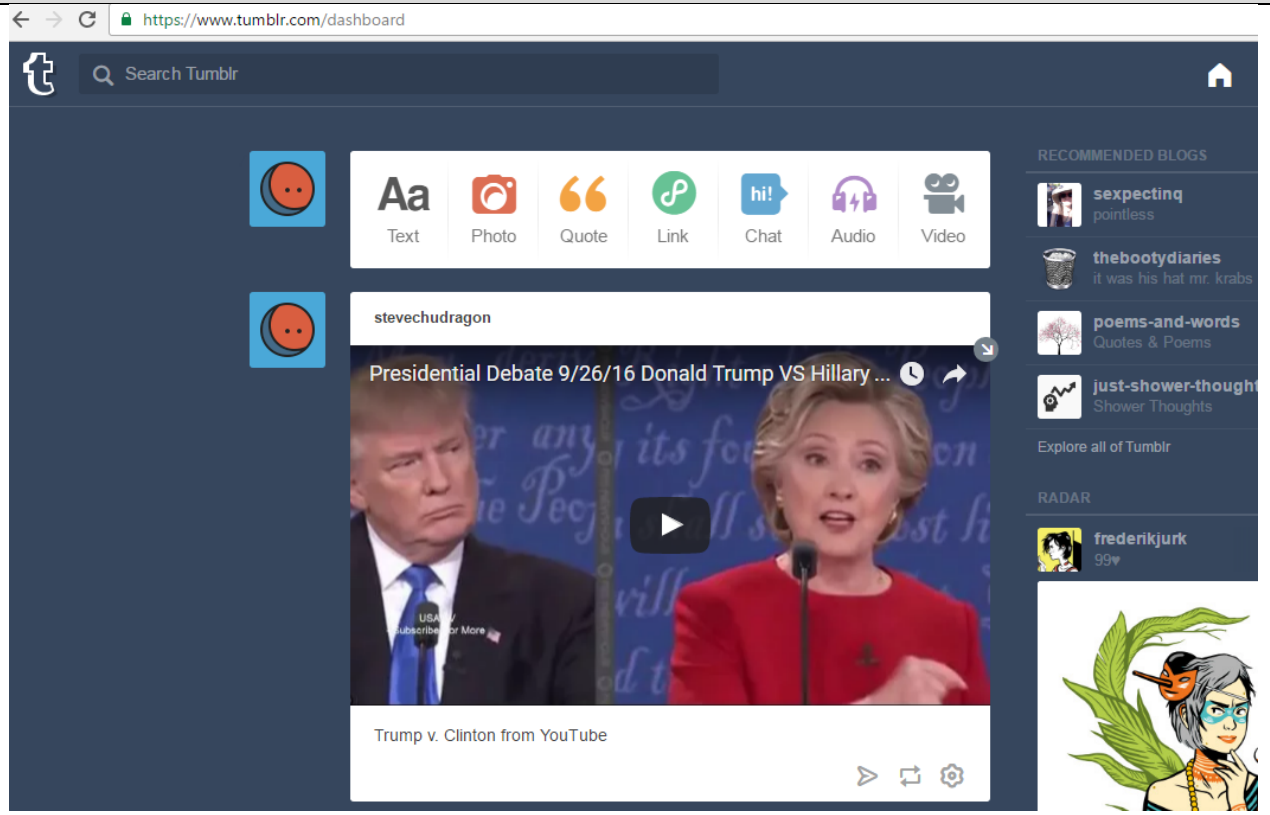
	
E4: cause displaying to a second user the stored metadata of the selected	As shown by Screen Shot 2 above, after the blog or reblog of a picture of a mountain road, the metadata of the picture is displayed on the second user (bethannyt)'s user interface, including graphic indicators of the blog. If desired, the second user can click the displayed metadata to access the full content of the blog.

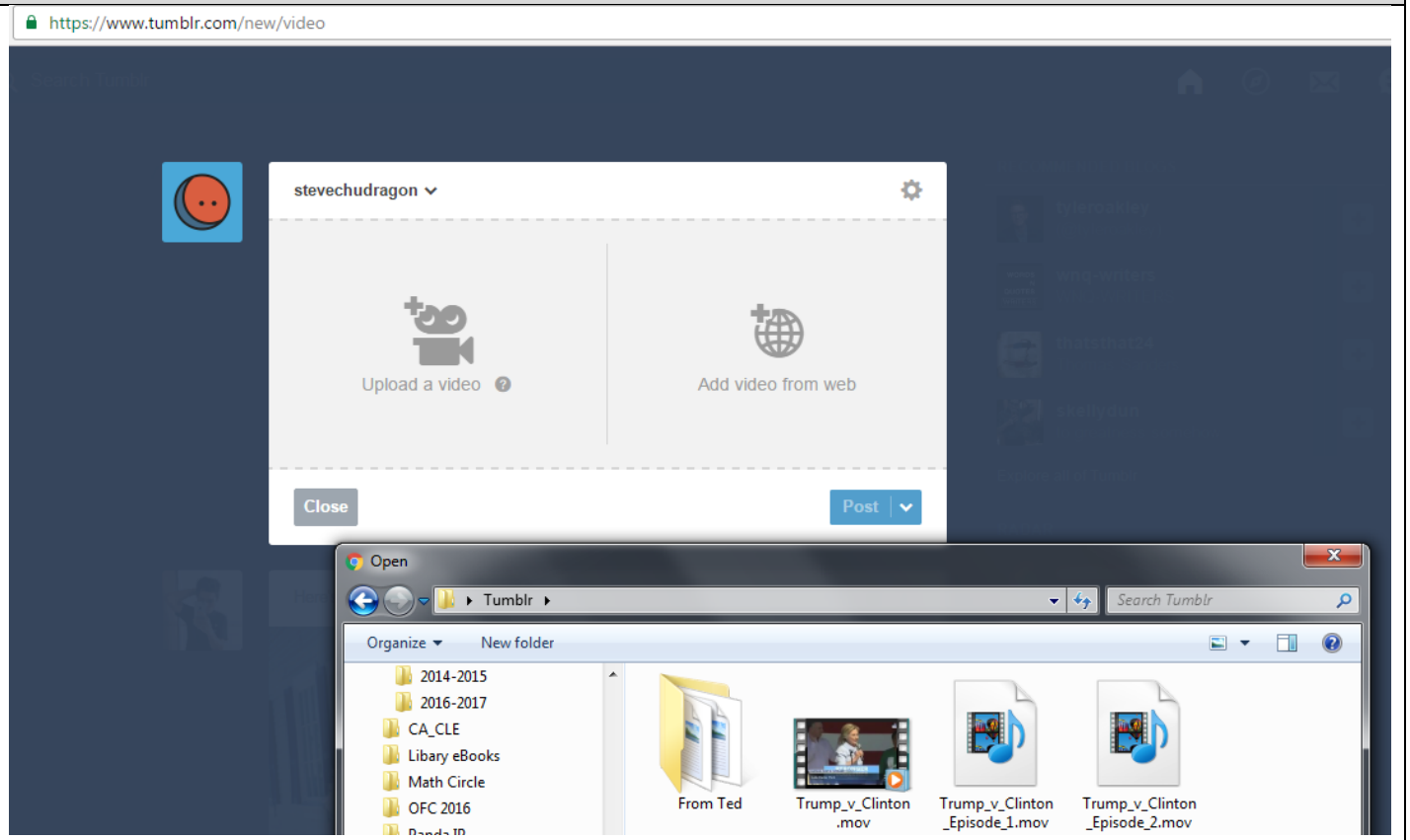
file or folder, including to display a graphic indicator of the selected file or folder, in the second user interface on a second end-user device to allow the second user access to the content of the selected file or folder through the stored metadata displayed in the second user interface.	

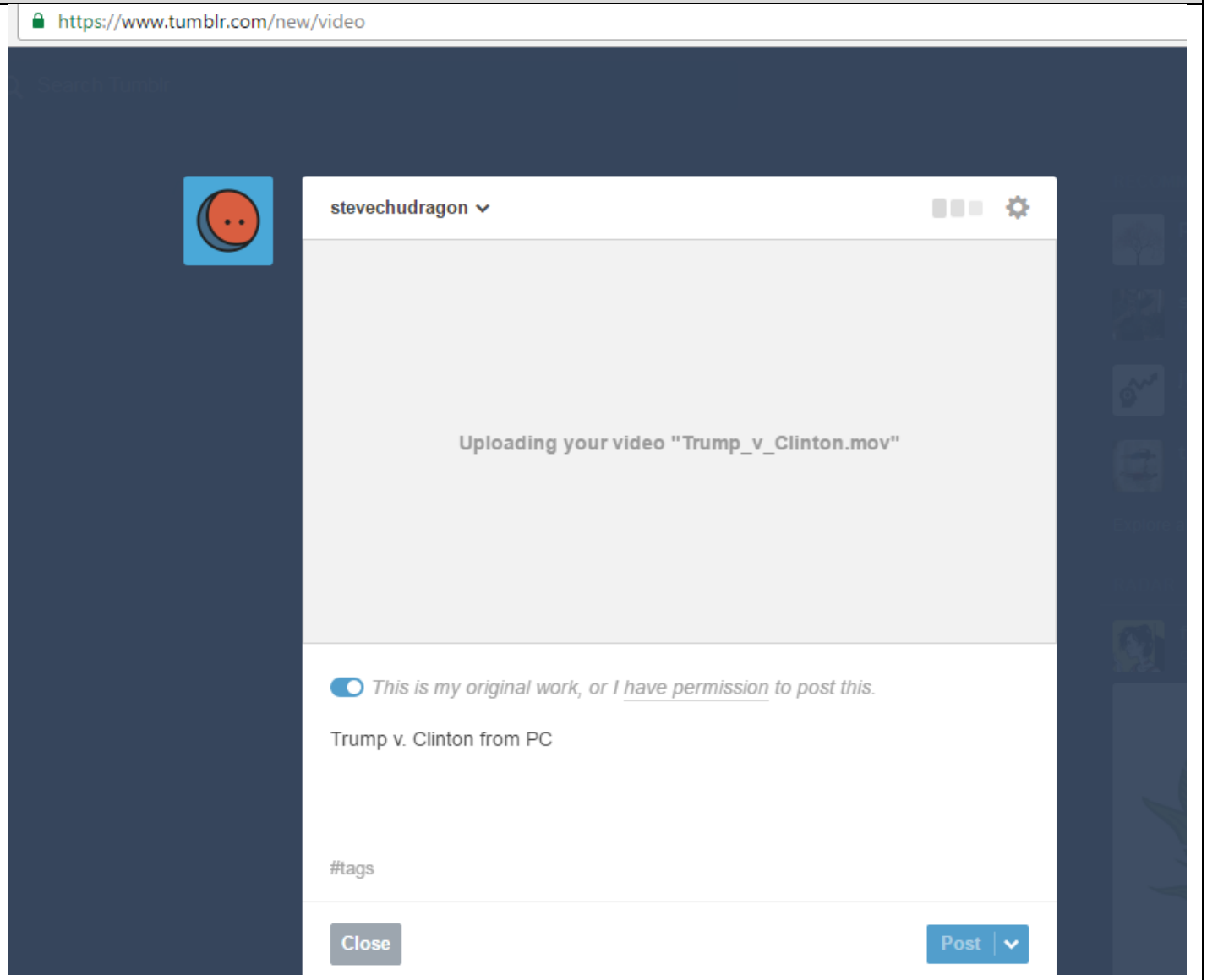
Exhibit H

Illustrative Claim Chart 3
U.S. Pat. No. 8,799,473

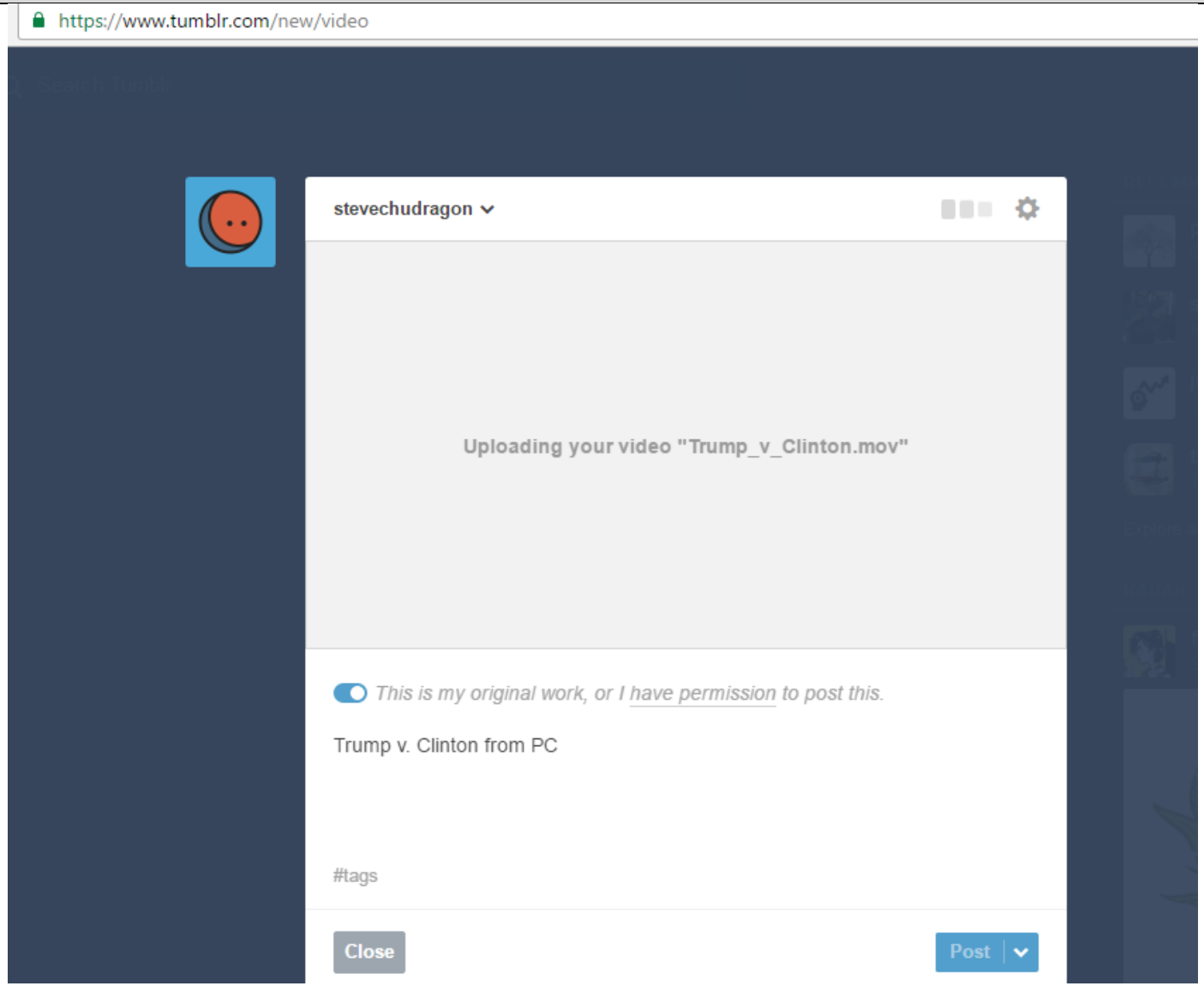
Claim 1	Yahoo's Tumblr Services ("Tumblr")
1. A server supporting access to resources, the server comprising:	Tumblr has a web server supporting access to resources including at least storage disk drive(s), folders, and content files.
at least one hardware processor, and	Tumblr's web server has at least one CPU hardware processor.
a network interface;	Tumblr's web server has at least one network interface for being connected by users from the Internet.
wherein the server is configure to perform followings:	Tumblr's web server is configured to perform the followings:
causing display of information about resources organized by the server in a web browser on a first end-user device;	Causing display of user interface information about resources such as a user ("stevechudragon")'s Tumblr home folder ("Dashboard", accessible at URL https://www.tumblr.com/dashboard) and content files organized by the server in a user's Chrome web browser on a personal computer of the user.

Claim 1	Yahoo's Tumblr Services ("Tumblr")
	
receiving a first request, for access a first resource, from the first end-user device upon a first user selecting the first resource from the information displayed on the first end-user device and submitting the first request;	When the user uploads a local file named "Trump_v_Clinton.mov" into his Tumblr home folder ("Dashboard") within the Tumblr user interface displayed in the web browser on the user's personal computer, Tumblr's web server receives a first "uploading a video" request, for accessing the user's Tumblr home folder from the user's personal computer upon the user selecting his Tumblr home folder from the user interface information displayed in the web browser on the user's personal computer and submitting the first "uploading a video" request.

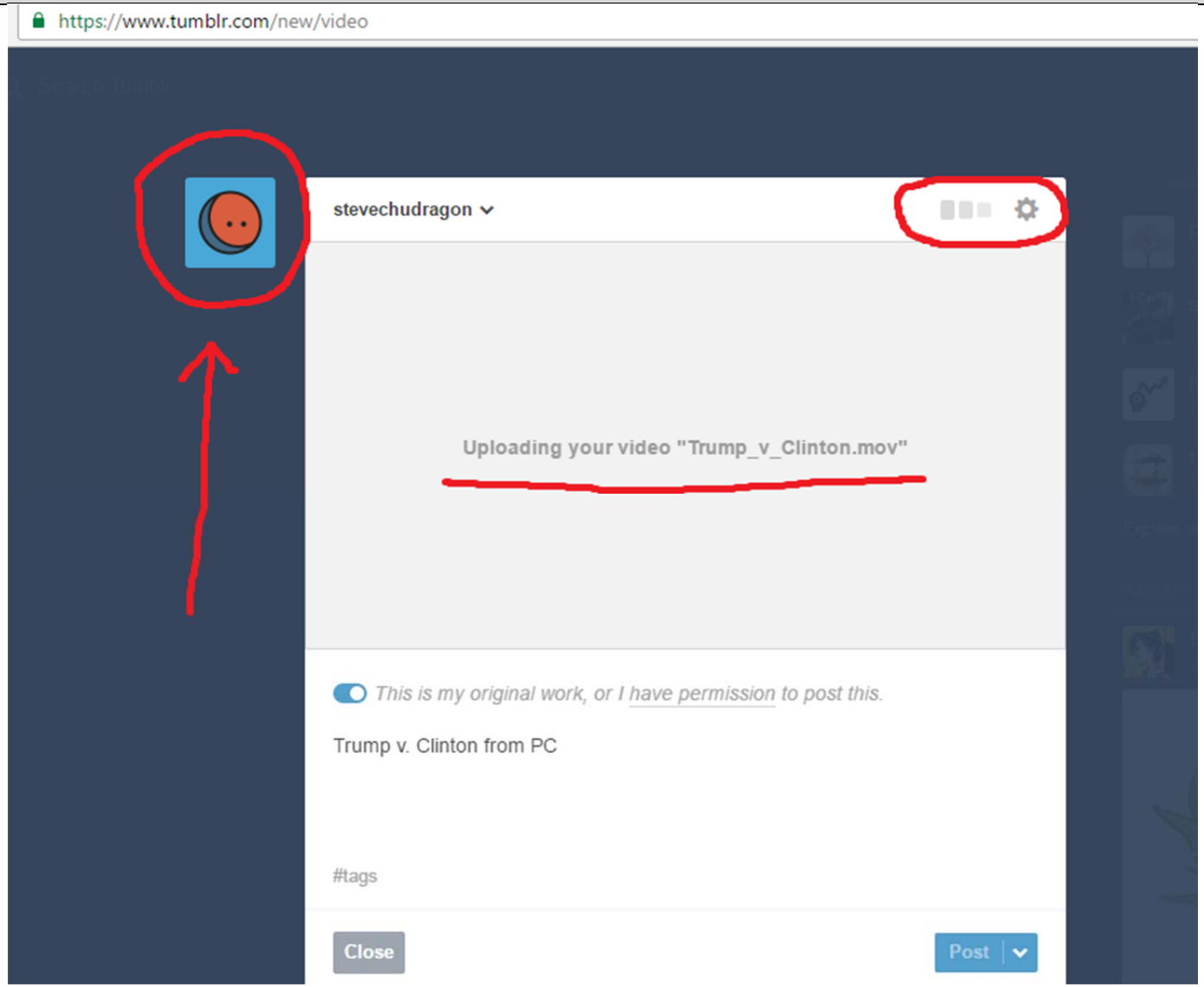
Claim 1	Yahoo's Tumblr Services ("Tumblr")
	

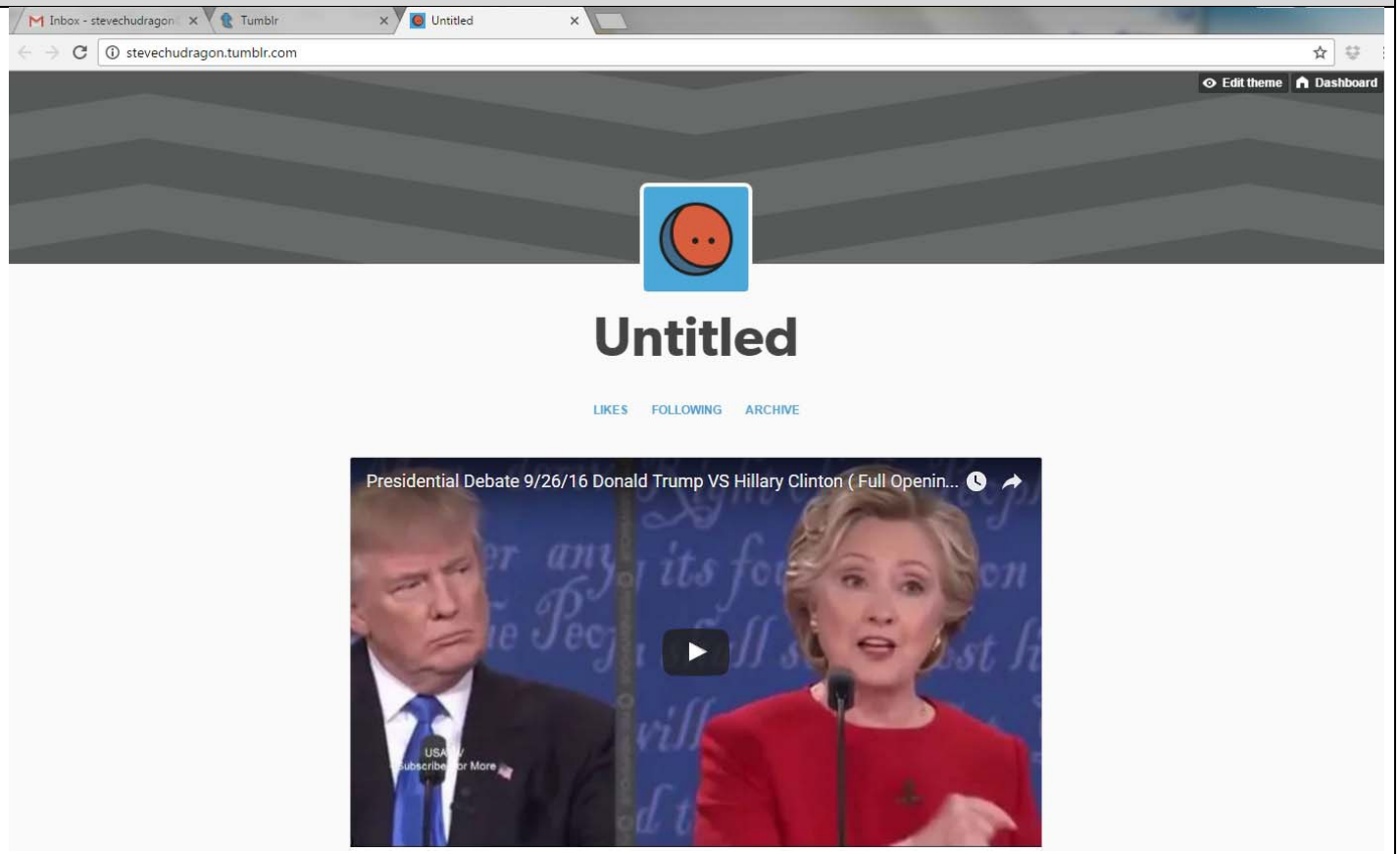
Claim 1	Yahoo's Tumblr Services ("Tumblr")
	

Claim 1	Yahoo's Tumblr Services ("Tumblr")
storing information about the first request and invoking a lock protection to protect the storing of the first request;	The Tumblr web server stores information about the user's first "uploading a video" request and invokes a lock protection to protect the storing of the user's first "uploading a video" request.

Claim 1	Yahoo's Tumblr Services ("Tumblr")
	

Claim 1	Yahoo's Tumblr Services ("Tumblr")
processing the first request, including to process the first request in the background and cause the display of the information about the resources without blocking in the web browser during a regular network traffic to allow the first user selecting a second resource from the information displayed on the first end-user device and submitting a second request for access to the second resource without waiting for the completion of the first request; and	While Tumblr's web server is processing the user's first "uploading a video" request, including to process the first "uploading a video" request in the background and cause the display of the information about the user's Tumblr home folder without blocking in the web browser during a regular network traffic, the Tumblr web server allows the user to select an "Untitled" folder (accessible at URL http://stevechudragon.tumblr.com/), from the user interface information displayed in the web browser on the personal computer, and to submit a second "accessing the 'Untitled' folder" request for access to the "Untitled" folder within the Tumblr user interface, without waiting for the completion of the first "uploading a video" request; and

Claim 1	Yahoo's Tumblr Services ("Tumblr")
	

Claim 1	Yahoo's Tumblr Services ("Tumblr")
	
deleting the stored information about the first request when the first request is completed.	After the first "uploading a video" request of uploading the local file "Trump_v_Clinton.mov" into the user's home folder is completed, the Tumblr web server deletes at least certain stored information about the first "uploading a video" request.

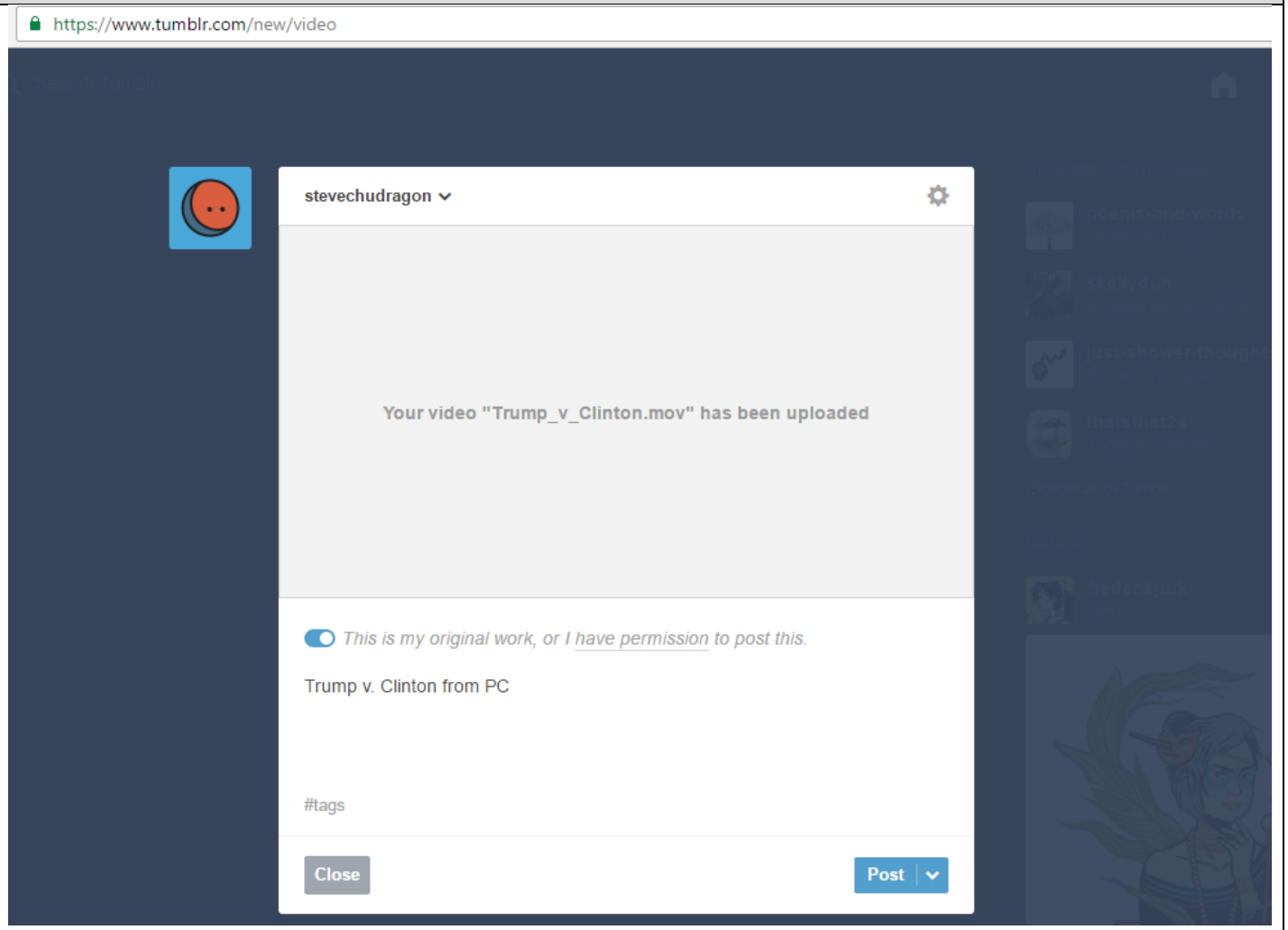
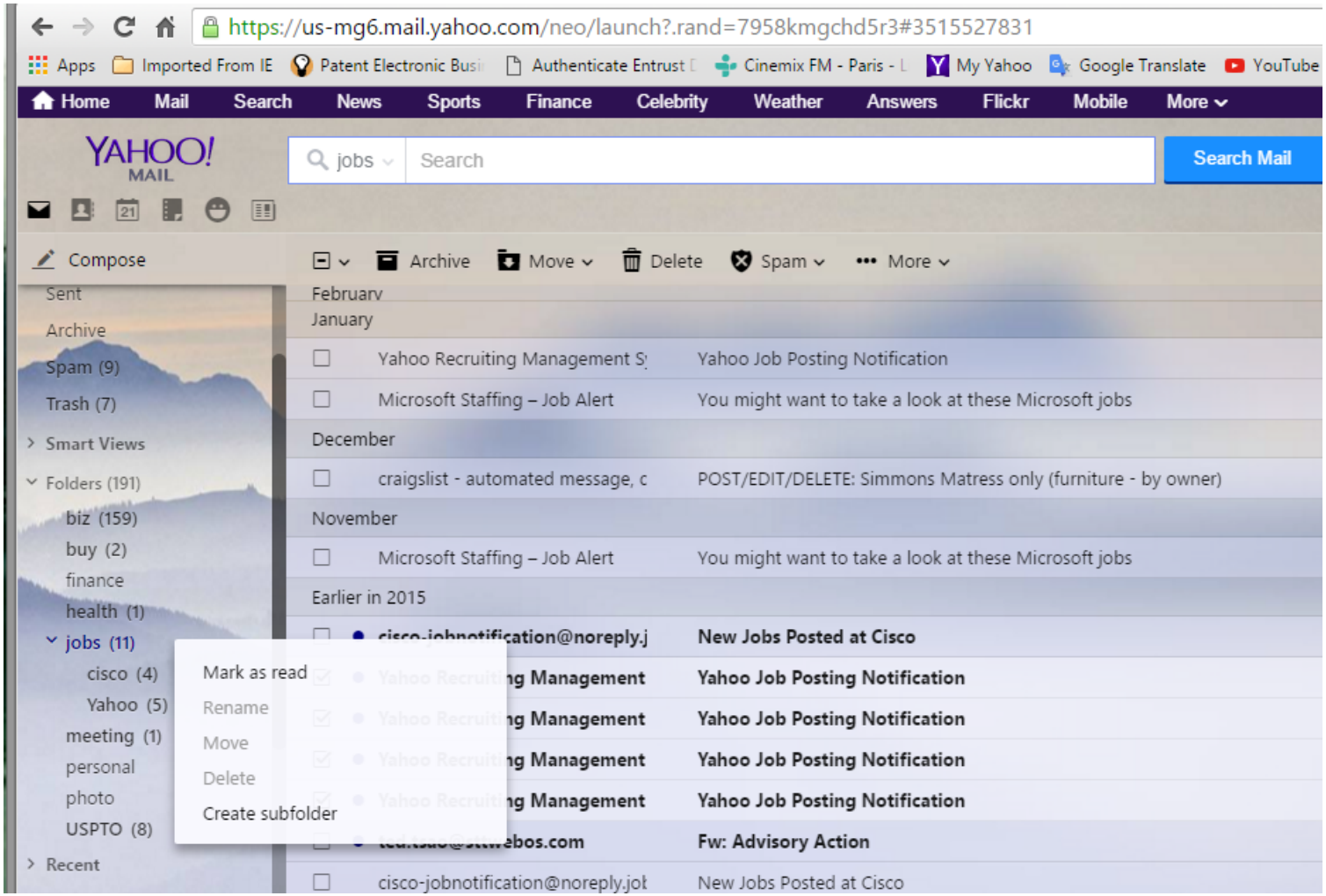
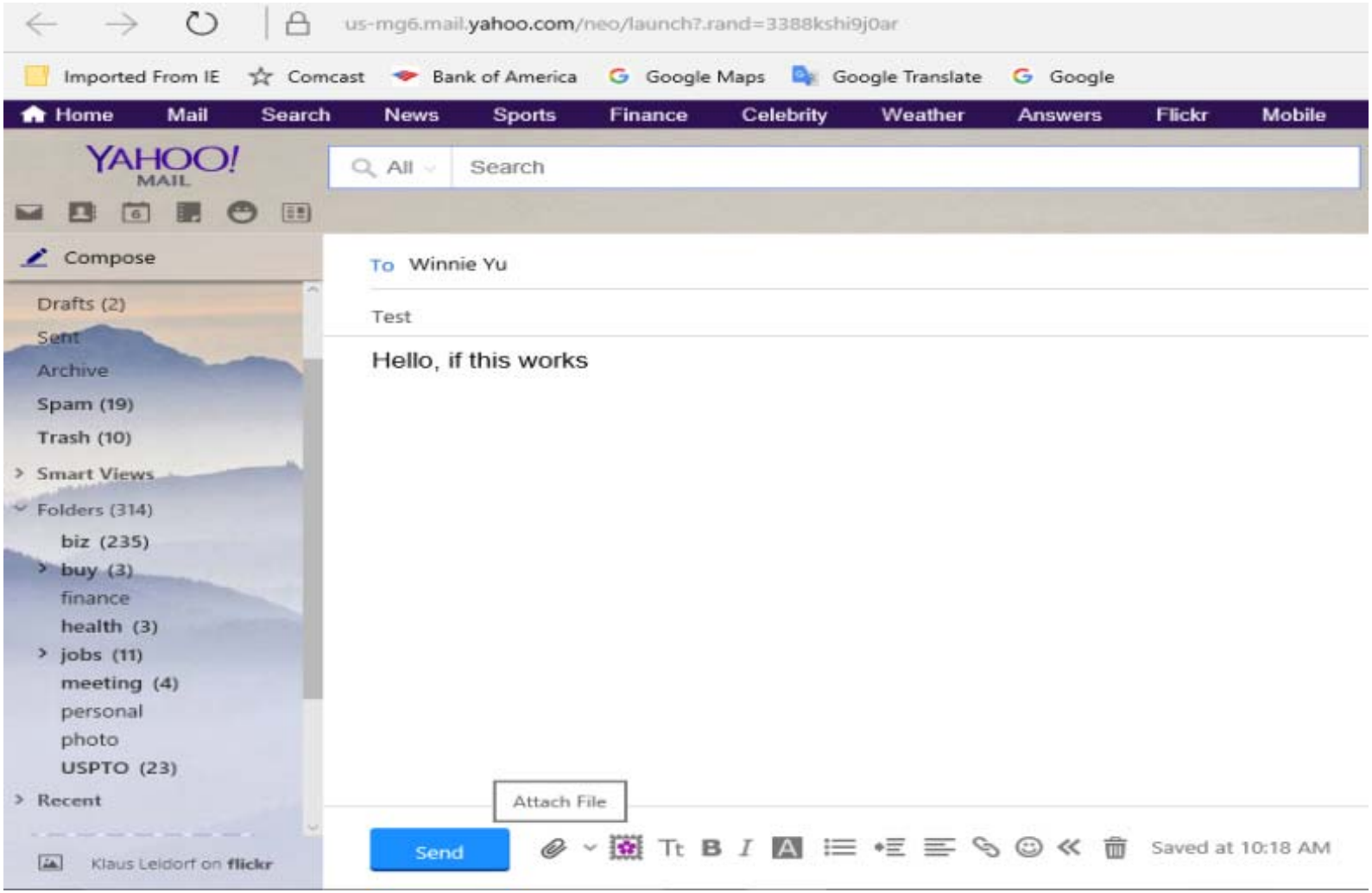
Claim 1	Yahoo's Tumblr Services ("Tumblr")
	

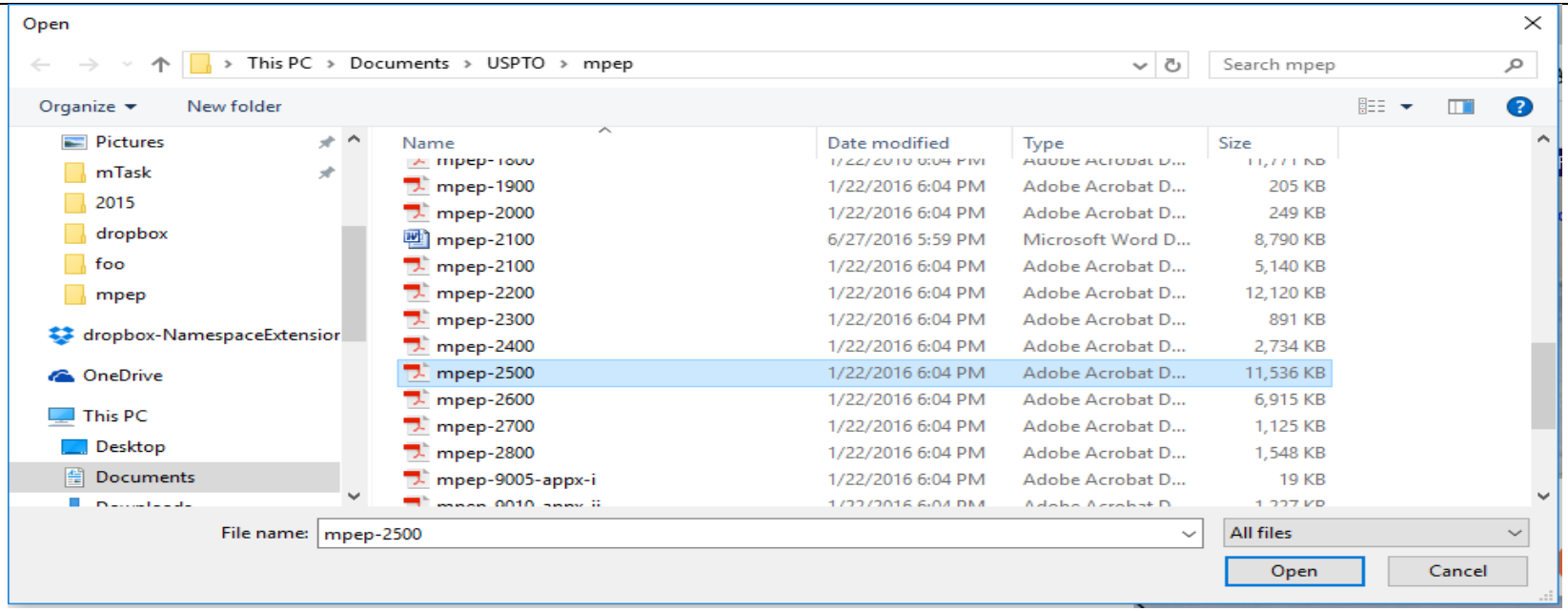
Exhibit I

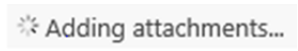
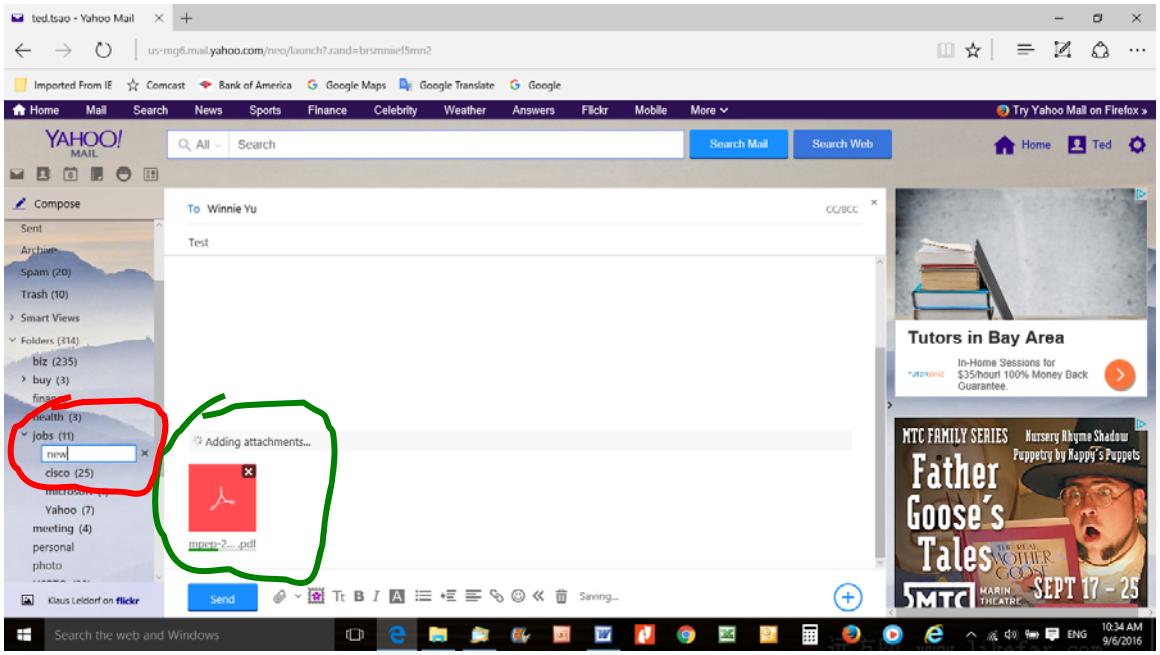
Illustrative Claim Chart 4
U.S. Pat. No. 8,799,473

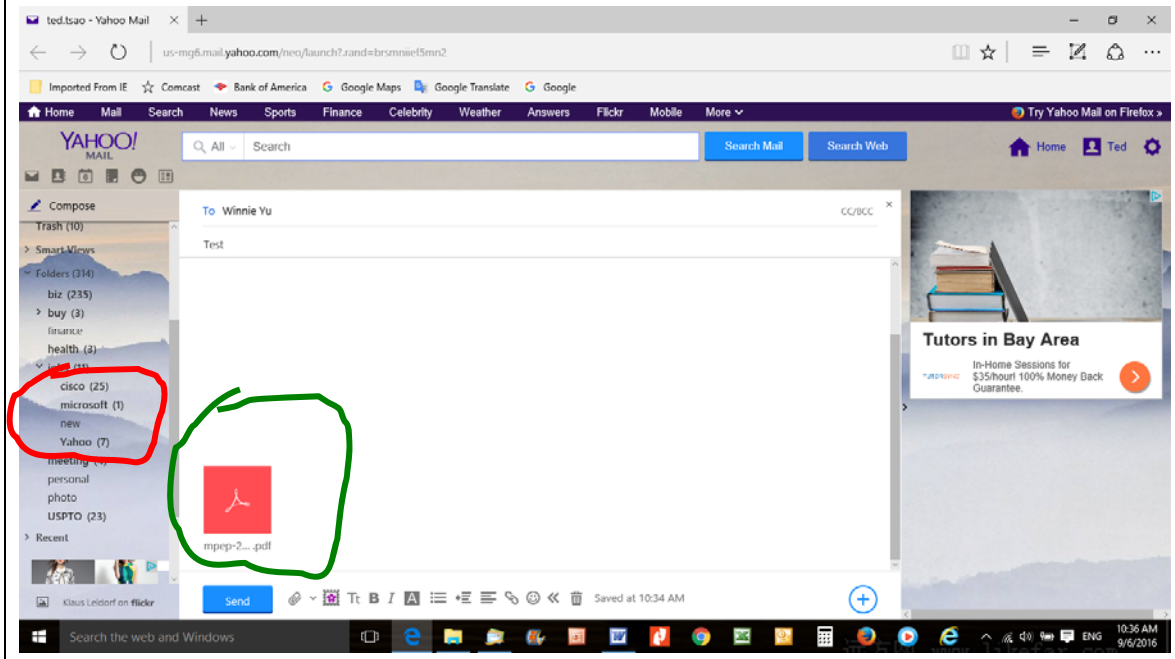
Claim 1	Yahoo Email Service ("Yahoo Email")
1. A server supporting access to resources, the server comprising:	A Yahoo Email server or servers (collectively "Yahoo Email Server" or "Email Server") provide user access to resources, such as message, files, web folder, operation menu etc.
at least one hardware processor, and a network interface; wherein the server is configure to perform followings:	The Yahoo Email Server, like any network servers, contains at least one hardware processor. Yahoo Email Server must also have a network interface for handling the email traffic through the Internet and other networks.
causing display of information about resources organized by	The Email Server sends information about resources such as information of email folders, email messages, operation menu, etc to a user's device for displaying in a browser of an device. As shown in the screenshot below, the "resources organized by the server in a web browser" include folders and subfolders that the user may access, emails stored in a folder or subfolder, operations that the user may do with the folders (e.g., to "rename" or "move" a folder as shown on the popup window), operations that the user may do with emails (e.g., to "archive," "move" or "delete" an email as shown on the menu on the top of the email list), and general task categories including "Compose" an email, look at the calendar or contact information etc.

the server in a web browser on a first end-user device;	
receiving a first request,	The screenshots below demonstrate one example of how Yahoo Email Server practices this element. In this example, the Email Server receives a “first request” to attach a file named “mpep-2500” to an email. This “first request” is made through the web browser environment

for access a first resource, from the first end-user device upon a first user selecting the first resource from the information displayed on the first end-user device and submitting the first request;	of a computer (i.e., "the first end-user device"), while the user is composing an email to "Winnie," by clicking the button "Attach File," as shown in the first screenshot below:  After the button "Attach File" is pressed, a popup window appears, as shown below:

	 By the user's selecting the file name "mpep-2500" from a file folder displayed in the popup window, as shown above, and then clicking "open" in the popup window, the "first request" for attaching the file mpep-2500 to the email to Winnie Yu is submitted to the Email Server by the end-user device, i.e., the computer.
storing information about the first request and invoking a lock protection to protect the storing of the first request;	The Email Server must store the user's request for attaching a file in the RAM and must invoke a lock protection for the stored information, when it processes the request. The "lock protection" here can be any process or thread synchronization mechanisms developed before the patent for preventing two processes from inferencing with each other. This lock protection must exist in the Yahoo Email Server because of the concurrent process to be discussed later in this claim chart.

processing the first request, including to process the first request in the background and cause the display of the information about the resources without blocking in the web browser during a regular network traffic to allow the first user selecting a second resource from the information displayed on the first end-user device and submitting a second request for access to	On the email interface, while the first request (i.e., attaching file, circled in green) is running in the background and the displaying of the information is not being blocked on the device, the user is able to submit the second request (e.g., creating a subfolder, circled in red) during the regular traffic without waiting for the completion of the first request. The second request is submitted by selecting a second source (i.e., the new folder icon) displayed on the user device. The screen printout below shows that while the new folder is being created, the first request (i.e., attaching a file) is still being processed as indicated by the icon  

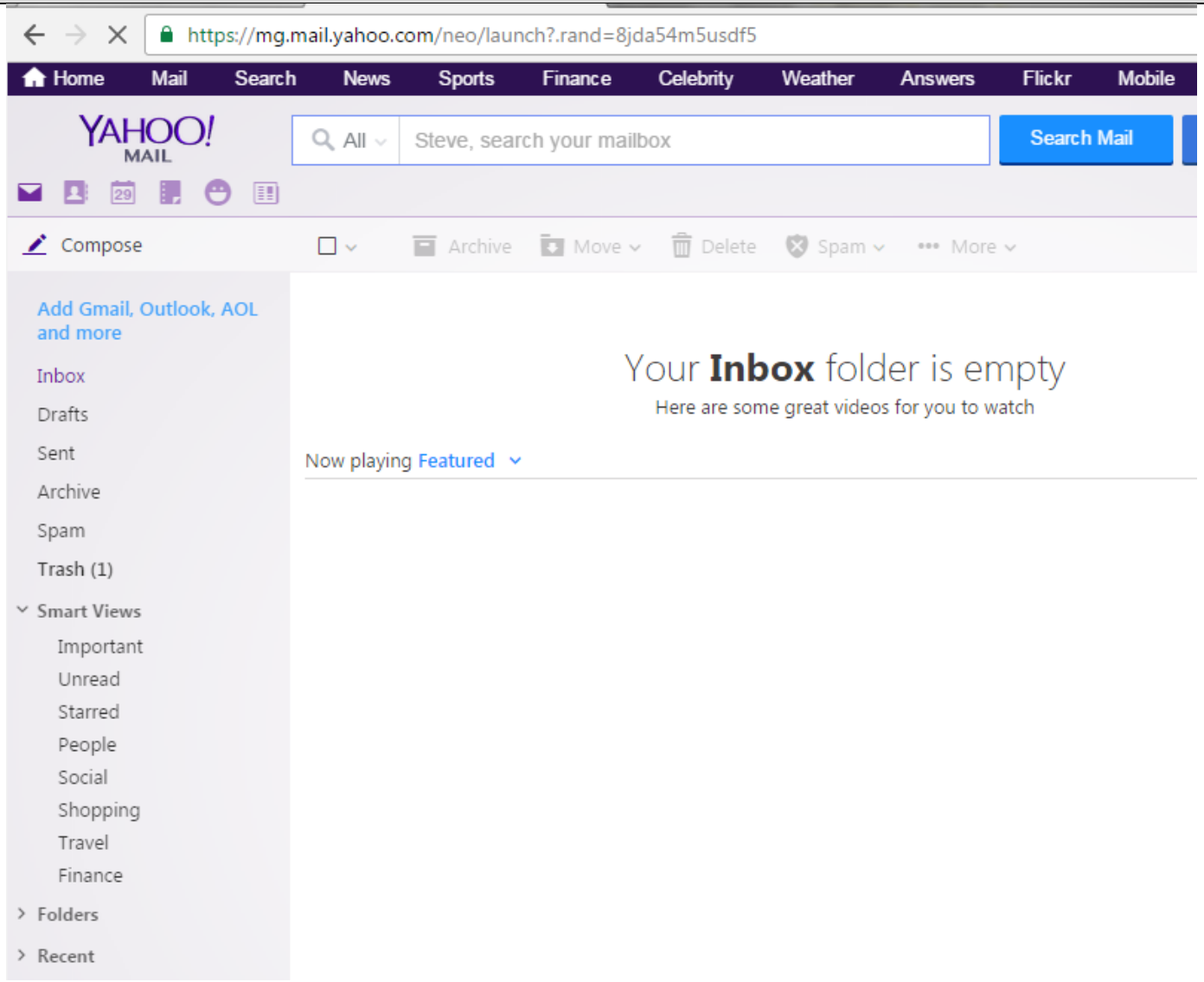
the second resource without waiting for the completion of the first request; and	The screen below shows when both the first and second requests are completed: 
Element 5: deleting the stored information about the first request when the first request	The email server must delete the first request information when the first request is completed if it is a reasonably efficient email server.

is completed.	

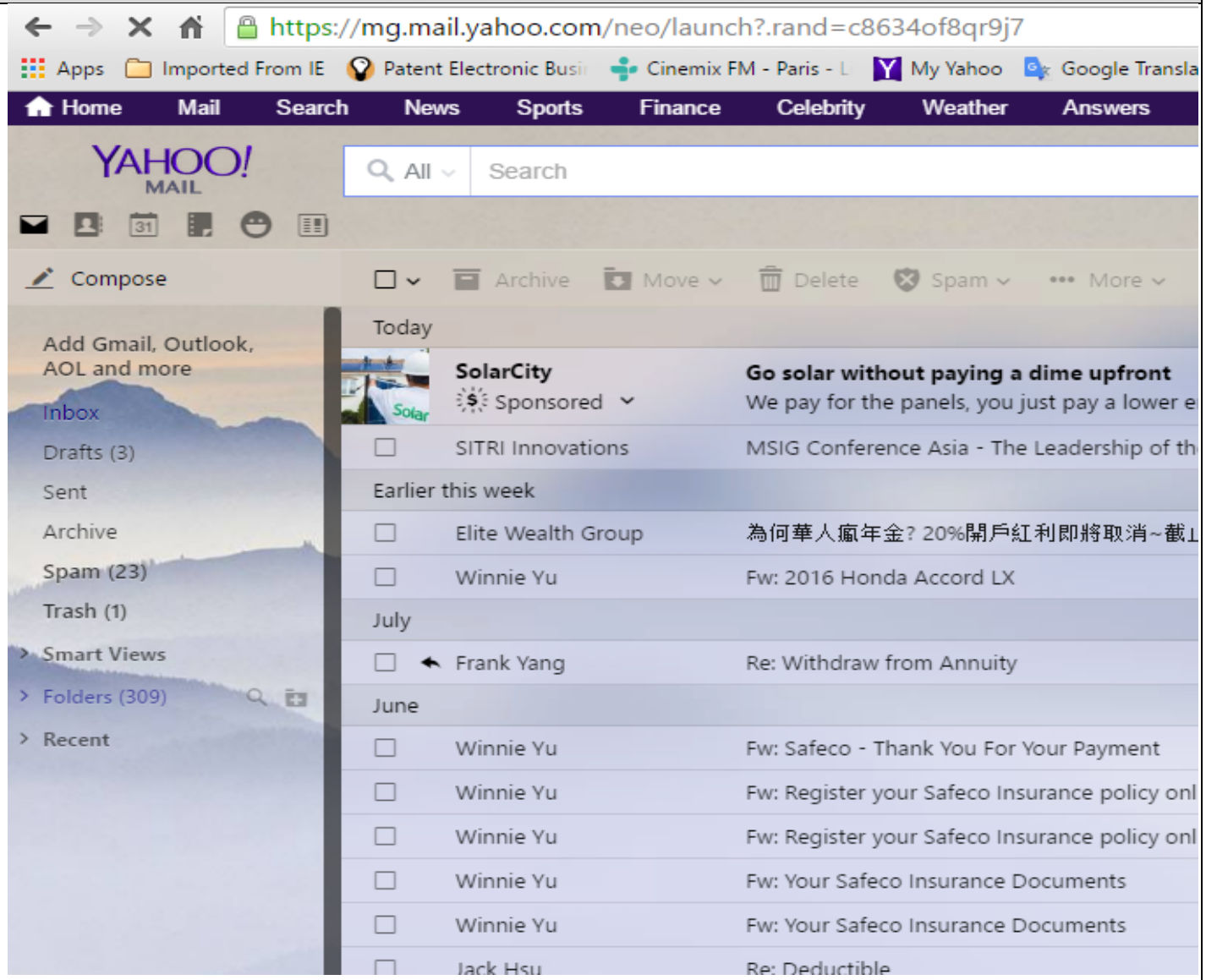
Exhibit J

Illustrative Claim Chart 5
U.S. Pat. No. 9,280,547

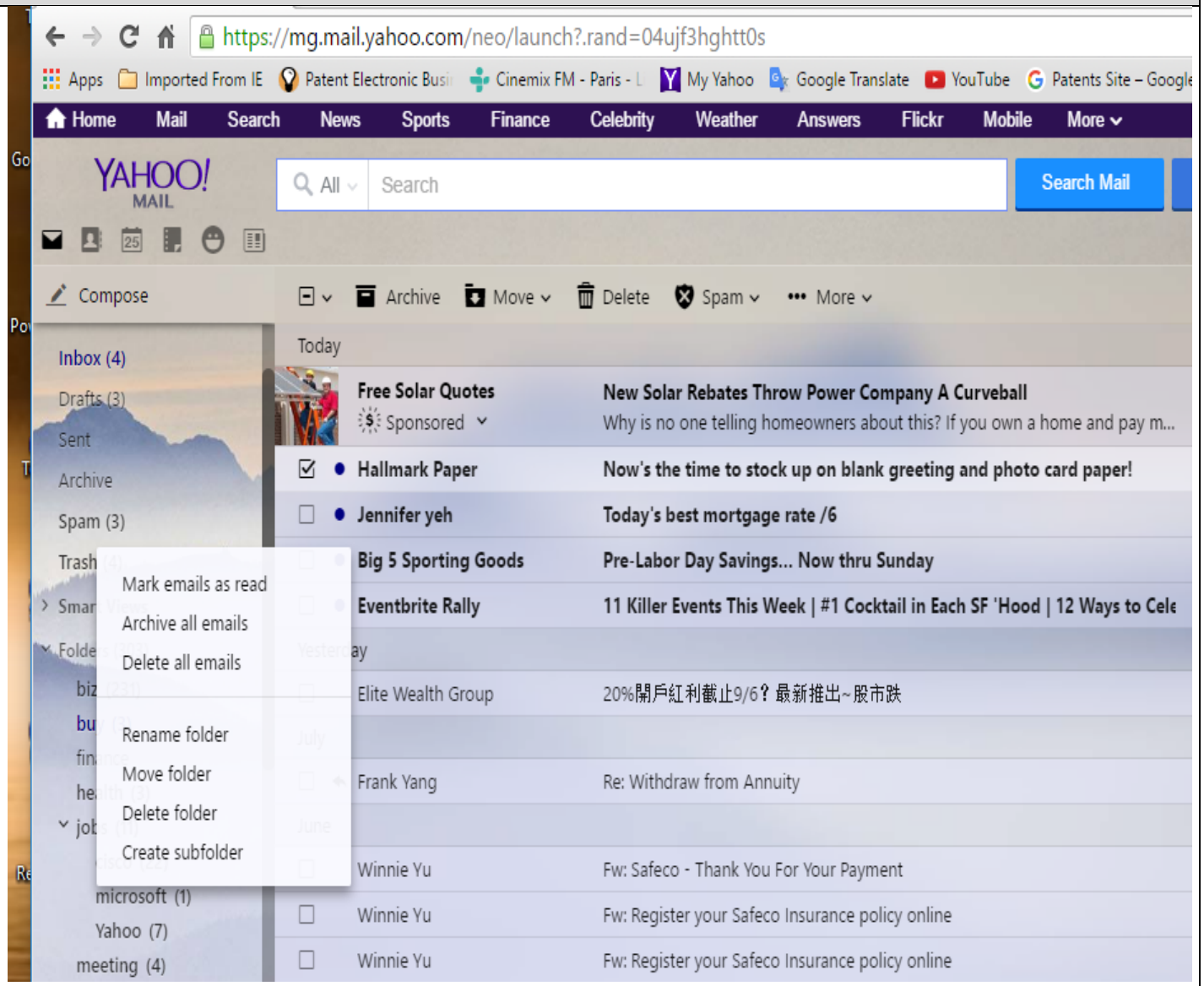
Claim	Yahoo's Mail Service Hardware and Software ("Yahoo Mail")
1. A server supporting a plurality of users access to remote folder structures, the server comprising:	Yahoo Mail has at least one email server supporting a plurality of users' access to remote folder structures located on Yahoo Mail's server or servers (collectively, the "Mail Server"). The screenshot below shows the main page of Yahoo Mail of a user named "Steve." Through the mail page, Steve may access a remote folder structure as shown on the left side of the screen shot. Obviously, Yahoo Mail allows "a plurality of users" to access their Yahoo Mail at the same time.

Claim	Yahoo's Mail Service Hardware and Software ("Yahoo Mail")
	

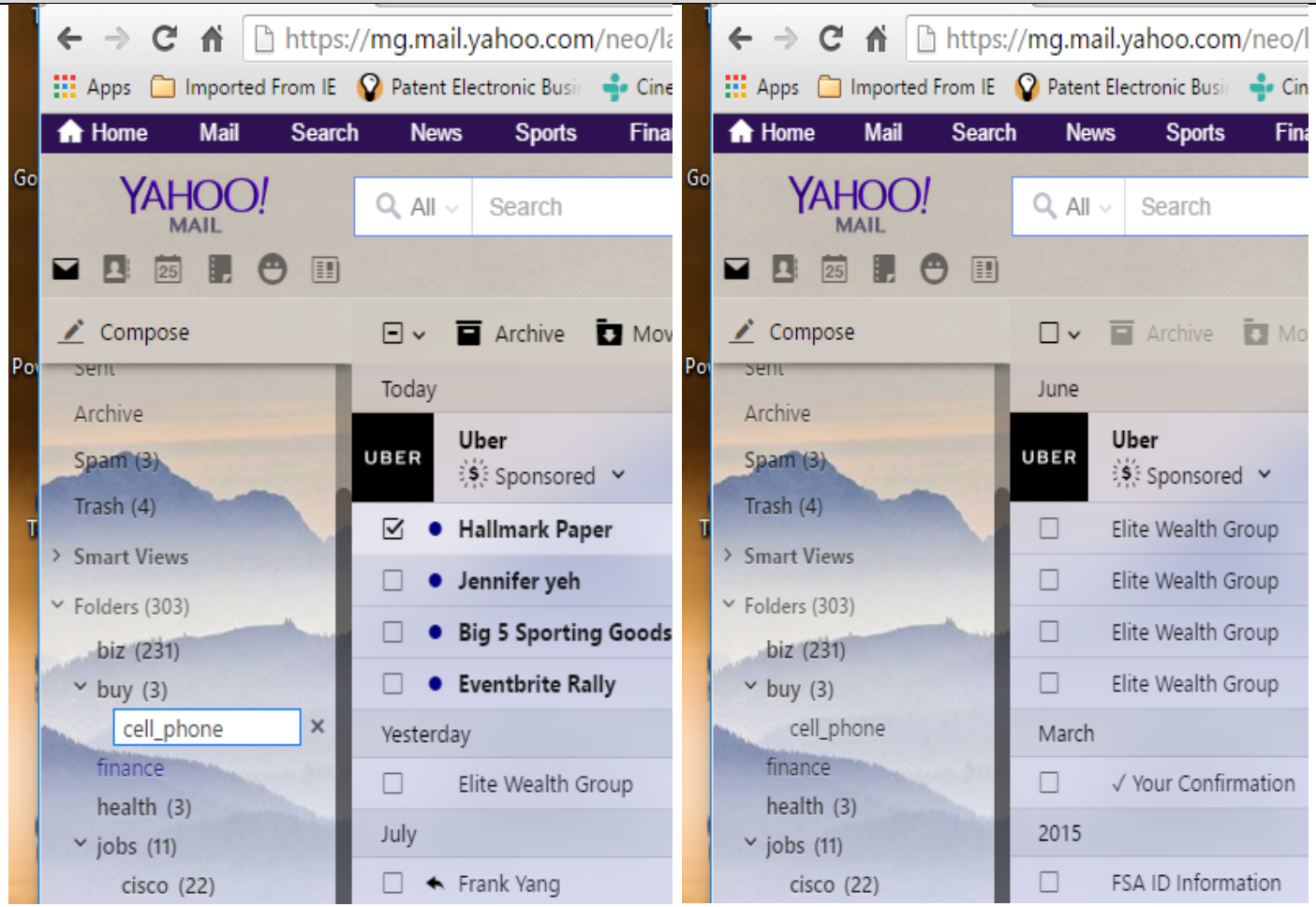
Claim	Yahoo's Mail Service Hardware and Software ("Yahoo Mail")
	The server comprises the following elements:
memory, and	The Mail Server is a computer that has memory.
non-transitory computer-readable medium comprising program code which, being executed by the server, configures the server to:	The Mail Server is a computer that has non-transitory computer-readable medium comprising program code which, being executed by the server, configures the server to accomplish the following tasks:
create a first per user-session hierarchical list in the memory for a user session initiated via a first end-user device by a first one of the users for access to a folder structure served by the server,	The Mail Server creates a first hierarchical list in the memory for a user session initiated by the user's computer to represent folder structure for the user's account during a "user session," e.g., when the user login onto his or her user account with Yahoo Mail using an end-user device such as a smartphone or laptop computer. The "first hierarchical list" is shown on the left side of the screenshot below, including folders named as, e.g., "Smart Views" and "Folders."

Claim	Yahoo's Mail Service Hardware and Software ("Yahoo Mail")
	

Claim	Yahoo's Mail Service Hardware and Software ("Yahoo Mail")
the first hierarchical list representing the folder structure in a reduced form,	The "first hierarchical list" in the above screen shot represents the Yahoo Mail folder structure in a reduced form, as it only shows the first level folders. Some folders (e.g., "Smart Views") have an arrow next to the folder names, indicating the hidden subfolders. The existence of hidden subfolders indicates that the "first hierarchical list" is a "folder structure in a reduced form."
the folder structure comprising one or more folders, where each of the one or more folders is used for holding at least one data object,	The Yahoo Mail folder structure shown above contains several folders that are used to hold data objects, which are in general email messages. For example, the screenshot above shows the "Folders" was holding "309" emails, a list of which is shown on the right side of the screenshot.
send a user interface comprising the first hierarchical list to the first end-user device to be displayed thereon,	The Mail Server sends to the user's device (e.g., a computer) a "user interface," for being displayed in a web browser running on the device. An example of such user interface is shown in the screenshot above, which includes "the first hierarchical list" as explained above.
the displayed first hierarchical list being navigated by the first one of the users to request access to the folder structure;	The user can certainly navigate the "first hierarchical list" to request access to the folder structure by, e.g., clicking a folder or by right clicking the mouse to show the maneuvers that can be done with the folders, such as renaming, deleting, or moving a folder or creating a subfolder. The following screenshot shows such task list for the folder list.

Claim	Yahoo's Mail Service Hardware and Software ("Yahoo Mail")
	

Claim	Yahoo's Mail Service Hardware and Software ("Yahoo Mail")
process the request for access to the folder structure received from the first end-user device,	The Mail Server of course processes the user's request for accessing the folder structure. For example, if the User clicks a folder on the folder list, the Mail Server will show the contents of the folder on the right side of the screen. If the user requests to rename a folder, the Mail Serve will rename the folder.
wherein the program code to process the request includes to update the folder structure, and also update the first hierarchical list in the memory to reflect the updated folder structure in accordance to the request,	The program code of the Mail Server that processes the user request includes the code for updating the folder structure and updating the "first hierarchical list" to reflect the changes to the folder structure. For example, the screen shots below show an example of how the program code of the Mail Server updates the folder structure and the "first hierarchical list". The screenshot on the left shows the user in the process of requesting the addition of a subfolder named "cell_phone" under the folder "buy," while the "first hierarchical list" is being updated to reflect the requested changes. The screenshot on the right shows the folder structure is updated and such update is preserved in the Mail Server, because when the user logon again later the changes in the folder structure (i.e., the addition of the "cell_phone" subfolder) is still there.

Claim	Yahoo's Mail Service Hardware and Software ("Yahoo Mail")
	
wherein the server sends an updated user interface comprising the updated first hierarchical list to the first end-user device to be	As shown by the screenshot on the right above, the Mail Server indeed sent an updated user interface with the "first hierarchical list" to the user's device, showing the addition of the new subfolder "cell_phone." When the user ends the session and exit the Mail Server, the server must delete the "first hierarchical list" from its memory or do something equivalent to prevent the "first hierarchical list" of the user being accessed or used by other users, and to make sure the next time the user logs on, a new "first hierarchical list" will be

Claim	Yahoo's Mail Service Hardware and Software ("Yahoo Mail")
displayed thereon during the user session, and deletes the first hierarchical list from the memory in response to exit of the user session.	generated according to the most updated folder structure.